IBM External Submission #24167

Reprint Courtesy of International Business Machines Corporation, © 2015 International Business Machines Corporation

INTERNATIONAL BUSINESS MACHINES CORPORATION (IBM) ARMONK, NEW YORK 10504

PERMISSION TO REPRINT IBM COPYRIGHTED PUBLICATIONS

Each reprint must be accompanied by the following credit line: **"Reprint Courtesy of International Business Machines Corporation, © (year) International Business Machines Corporation".** The credit line normally should appear on the page where the reprint appears, either under the title or as a footnote.

If the foregoing is inconvenient, the credit line may be placed on the face or back of the title page (or front cover, if there is no title page) or in a conveniently viewable manner with suitable reference to the place where the reprint appears.

When multiple IBM materials are reprinted in the same publication, a consolidated credit paragraph may be used on the title page, or in a conveniently viewable manner listing the titles, corresponding copyright notices and references to the points where the reprints appear.

It is the understanding of **International Business Machines Corporation** that the purpose for which its material is being used is accurate and true as stated in the original request.

Permission to quote from, transmit electronically or reprint IBM material is limited to the purpose and quantities originally requested and must not be construed as a blanket license to use the material for other purposes or to reprint other IBM copyrighted material.

IBM reserves the right to withdraw permission to use copyrighted material whenever, in its discretion, it feels that the privilege of using its material is being used in a way detrimental to its interest or the above instructions are not being followed properly to protect its copyright.

No permission is granted to use trademarks of **International Business Machines Corporation** and its affiliates apart from the incidental appearance of such trademarks in the titles, text, and illustrations of the named publications. Any proposed use of trademarks apart from such incidental appearance requires separate approval in writing and ordinarily cannot be given. The use of any IBM trademark should not be of a manner which might cause confusion of origin or appear to endorse non-IBM products.

THIS PERMISSION IS PROVIDED WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

INTERNATIONAL BUSINESS MACHINES CORPORATION

Dated: August 19, 2015

Workplace Shell Programming Reference

IBM OS/2 Toolkit Documentation

What's New

Following are the new classes and methods for OS/2 Warp Version 4.

New Classes

The following classes are new for OS/2 Warp Version 4. These include the Internet and Plug and Play classes. There is also a new WPImageFile class that is a parent to WPBitmap.

The new WPClassManager is responsible for modifying the SOM class manager to support object class replacement.

The following are the Workplace Shell Plug and Play classes.

- WPDevAudio
- WPDevBus
- WPDevCDRom
- WPDevCPU
- WPDevice
- WPDevDiskette
- WPDevDisplay
- WPDevHarddrive
- WPDevKeyboard
- WPDevMemory
- WPDevMouse
- WPDevParallel
- WPDevPeriph
- WPDevSerial
- WPDevTape
- WPDevTimer
- WPHwManager

The following classes are Workplace Shell Internet classes.

- WPHost
- WPHtml
- WPUrl
- WPUrlFolder

The following are other new Workplace Shell classes.

- WPClassManager
- WPImageFile

New Methods

The following Workplace Shell methods are new for OS/2 Warp Version 4.

There are no methods for the Internet classes.

Method Name	Description
wpGetTrueClassName	Returns the class name for the given object.

Method Name	Description
wpAddBecomePage	Adds the standard type page to the object's Settings notebook. This method should always be overridden in order to replace or remove the Become page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.
wpQueryHandleFromContents	Returns a handle to a known system data type based on a file's contents. This is an abstract method; subclasses that describe files for which a system handle type exist (such as WPIcon or WPBitmap) should subclass this method.
wpQueryPointerFromContents	Returns a pointer to shared memory containing the contents of the WPDataFile object.
wpRenderFromClipboard	Opens the clipboard and retrieves the type of data specified in the clipboard format passed in, then writes the clipboard data to a file.
wpSetContentsFromHandle	Sets a file's contents based on a handle to a known system data type. This is an abstract method; subclasses that describe files for which a system handle type exist (such as WPIcon or WPBitmap) should subclass this method.
wpSetContentsFromPointer	Sets a WPDataFile object's contents based on a pointer to shared memory.
wpclsAddClipboardAssoc	Adds an association between a clipboard format and a particular class name.
wpclsEnumClipboardClasses	Enumerates through all the object classes that a particular rendering mechanism supports.

Method Name	Description
wpQueryAutoLockup	Returns the current value of the automatic lockup option.
wpQueryDefaultDropOp	Queries the default drop operation of objects that are dropped on the Desktop. The default is set on the Desktop page of the Desktop's Settings notebook. This method can be overridden to control the Desktop's default drop operation

wpQueryLockupAutoDim	Returns the current value of the lockup auto-dim option.
wpQueryLockupBackground	Returns the current values for the lockup background.
wpQueryLockupFullScreen	Returns the current value of the lockup full-screen option.
wpQueryLockupOnStartup	Returns the current value of the lockup on the startup option.
wpQueryLockupTimeout	Returns the current timeout value for automatic lockup.
wpSetAutoLockup	Sets the value of the automatic lockup option.
wpSetDefaultDropOp	Sets the default drop operation of objects that are dropped on the Desktop. The default is set on the Desktop page of the Desktop's Settings notebook.
wpSetLockupAutoDim	Sets the value of the lockup auto-dim option.
wpSetLockupBackground	Sets the current values for the lockup background.
wpSetLockupFullScreen	Sets the value of the lockup full-screen option.
wpSetLockupOnStartup	Sets the value of the lockup on the startup option.
wpSetLockupTimeout	Sets the timeout value for automatic lockup.

Method Name	Description
wpAddSnoop1Page	Adds the Detection Level page to the settings notebook.

Method Name	Description
wpAddDeviceDevice1Page	Adds the first Device page to the settings notebook.
wpAddDeviceDevice2Page	Adds the second Device page to the settings notebook.
wpAddDeviceDMA1Page	Adds the DMA Channel page to the settings notebook.
wpAddDeviceIO1Page	Adds the I/O Ports page to the settings notebook.
wpAddDeviceIRQ1Page	Adds the IRQ Level page to the settings notebook.
wpAddDeviceMemory1Page	Adds the Memory page to the settings notebook.

Method Name	Description
wpAddReserveDriveLetterPage	Adds the reserve drive letter

	page to the Settings notebook. This method should always be overridden in order to replace or remove the Reserve Drive Letter page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

Method Name	Description
wpAddUserItemsToPopupMenu	Adds user-defined items to the popup menu.
wpQueryEASupport	Determines whether extended attributes are supported for the specified file system object.
wpQueryRefreshFlags	Queries the refresh flags for a file system object. The refresh flags consist of a DIRTYBIT and a FOUNDBIT that are used to allow refresh to detect deleted files and new files after asking a folder to re-populate.
wpSetRefreshFlags	Sets the refresh flags for a file system object. The refresh flags consist of a DIRTYBIT and a FOUNDBIT that are used to allow refresh to detect deleted files and new files, after asking a folder to re-populate.
wpclsQueryInstanceClass	Returns the name of a class that this object can become.

Method Name	Description
wpAddFolderMenu2Page	Adds the standard Menu2 page to the Settings notebook. This method should always be overridden in order to replace or remove the Menu page 2 from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.
wpFlushNotifications	Forces all pending file system notifications for a folder to be processed immediately.
wpFreeIconPosData	Frees the .ICONPOS extended

	attribute (EA) information that is allocated by wpInitIconPosData.
wpIdentify	Returns a string that is unique within a given container that identifies the specified object.
wpInitIconPosData	Reads the .ICONPOS extended attribute (EA) for the folder directory to determine where to position the object icons within the folder.
wpQueryFldrBackground	Returns the current values for the folder background.
wpQueryIconPosition	Finds the saved icon postion for this record.
wpQueryIconTextBackgroundColor	Returns the current color being used for the icon text background.
wpQueryIconTextColor	Returns the current color being used for the icon text in the specified view.
wpQueryIconTextVisibility	Returns the current state of icon text visibility for the specified view.
wpQueryMenuBarVisibility	Returns the visibility state of the folder's menu bar.
wpQueryOpenFolders	Allows for the enumeration of all open folders in the system.
wpQueryShadowTextColor	Returns the current color being used for the shadow text in the specified view.
wpRestoreFldrRunObjs	For open work area folders, goes through the list of objects last saved by wpStoreFldrRunObjs and reopens all views of those objects that were open when the work area folder was last closed.
wpSetFldrBackground	Sets the values for the folder background.
wpSetIconTextBackgroundColor	Sets the color to be used for the icon text background.
wpSetIconTextColor	Sets the color to be used for the icon text for the specified view.
wpSetIconTextVisibility	Sets the icon text visibility for the specified view.
wpSetMenuBarVisibility	Sets the visibility state of the folder's menu bar.
wpSetShadowTextColor	Sets the color to be used for the shadow text for the specified view.
wpStoreFldrRunObjs	Called when a work area folder is closed. Saves a list of the objects within the work area folder that are open when the work area folder is closed, then closes all open views of these objects.
wpStoreIconPosData	Stores the ICONPOS data passed to this method on the folder object's .ICONPOS extended

attribute (EA).

Method Name	Description
wpQueryBitmapData	Returns a pointer to the bitmap data for the image file.
wpQueryBitmapHandle	Returns a handle to the bitmap that can be displayed showing the contents of the image file and a handle to the palette to be used when displaying the bitmap.
wpQueryBitmapInfoHeader	Returns a pointer to the bitmap information header for the image file.
wpReadImageFile	Reads the image data for the image file and stores it in the instance data for the object. If the image data has already been read, this method simply returns TRUE. It does not refresh the image data.
wpSetBitmapData	Sets the bitmap data for the image file. This method will convert the bitmap data to the appropriate format and update the actual image data file.
wpWriteImageFile	Writes the image data for the image file from the instance data.

Method Name	Description
wpMoveJobObject	Moves a print job to a different printer.

Method Name	Description
wpAppendObject	Appends a file system object to another file system object.
wpAssertObjectMutexSem	This instance method verifies that an object's mutex semaphore is held for the current thread.
wpFilterMenu	Filters out options from the object's pop-up menu that do not apply. (OS/2 Version 4 or higher.)
wpFindTaskRec	An object is sent this message to return information specific to the task being performed in a task thread.
wpInsertMenuItems	Inserts items into the object's pop-up menu. (OS/2 Version 4 or higher.)
wpModifyMenu	Adds new options to the object's pop-up menu. (OS/2 Version 4 or higher.)

wpQueryMenuStyle	Returns the current menu style, which is either long or short.
wpQueryTrueStyle	Queries this object's current class true style.
wpReleaseObjectMutexSem	Releases the mutex semaphore for an object. Each object has associated with it a mutex semaphore that can be used to serialize access to resources.
wpReplaceObject	Replaces an object in another folder.
wpRequestObjectMutexSem	Requests the mutex semaphore for an object. Each object has associated with it a mutex semaphore that can be used to serialize access to resources.
wpSetFolder	Sets this object to reside in the specified folder.
wpSetMenuStyle	Sets the menu style to be either long or short.
wpSetTaskRec	Adds, replaces, or deletes a task data block.

Method Name	Description
wpFreeCellData	Frees the data for the specified cell in a palette object.

Method Name	Description
wpSetJobProperties	Sets the print job properties.

Method Name	Description
wpQueryWarnBeforeStart (WPProgram)	Queries the program reference's Warn Before Start flag. If this flag is set, put up a warning dialog before starting the program.
wpQueryWarnBeforeStart (WPProgramFile)	Queries the program reference's Warn Before Start flag. If this flag is set, put up a warning dialog before starting the program.
wpSetWarnBeforeStart (WPProgram)	Sets the program reference's Warn Before Start flag. If this flag is set, put up a warning dialog before starting the program.
wpSetWarnBeforeStart (WPProgramFile)	Sets the program reference's Warn Before Start flag. If this flag is set, put up a warning dialog before starting the program.

Method Name	Description
wpAddSoundSchemePage	Adds the Sound Scheme page to the Settings notebook. This method should always be overridden in order to replace or remove the Sound Scheme page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

Method Name	Description
wpAddSetPortRedirPage	Adds the Set Port Redirection page to the Settings notebook. This method should always be overridden in order to replace or remove the Port Redirection page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.
wpAddSetSyncPage	Adds the Set Synchronous page to the Settings notebook. This method should always be overridden in order to replace or remove the Sync page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.
wpAddSetViewPage	Adds the Set View page to the Settings notebook. This method should always be overridden in order to replace or remove the Set View page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

Method Name	Description
wpAddSysFdrMenuPage	Adds a page to the Sytem Settings notebook to enable and disable OpenDoc menu options. This method should always be overridden in order to replace or remove the Folder Menu page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.
wpAddSystem256ColorPage	Adds the 256 Color page to the System Settings notebook. This method should always be overridden in order to replace or remove the 256 Color page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

New Presentation Manager Functions

Method Name	Description
WinNotebookButtonFromID	This function exports a convience function to find BS_NOTEBOOKBUTTON buttons. WinNotebookButtonFromID, given the HWND of the dialog of the properties page and the ID of the button along with the button of the properties page (for example, Undo, Default, or Help), returns the HWND for that button. This is used with the buttons of the BS_NOTEBOOKBUTTON style. Ownership of buttons of this style is changed from the dialog to the notebook after the dialog is loaded.
WinWaitForShell	This function allows applications to wait for the Shell to be available.

```
WinWaitForShell is used by an
application when it needs to
wait for the Workplace Shell
to be available in order for
the application to continue.
For example, the application
might be started in
STARTUP.CMD, and yet it wants
to create objects on the
Workplace Shell.  The
application can use
WinWaitForShell to wait for
the Workplace Shell to be
ready to start receiving the
create requests. By using the
optional query bit, it can
check on those conditions
without waiting on them.
```

How to Use this Book

This reference is a detailed technical guide and reference for application programmers. It gives reference information and code examples to enable you to write source code using Workplace classes and methods.

Before you begin to use this information, it would be helpful to understand how you can:

- Expand the Contents to see all available topics
- Obtain additional information for a highlighted word or phrase
- Use action bar choices
- Use the programming information.

How to Use the Contents

When the Contents window first appears, some topics have a plus (+) sign beside them. The plus sign indicates that additional topics are available.

To expand the Contents if you are using a mouse, click on the plus sign. If you are using the keyboard, use the Up or Down Arrow key to highlight the topic, and press the plus (+) key. For example, **Code Pages** has a plus sign beside it. To see additional topics for that heading, click on the plus sign or highlight that topic and press the plus (+) key.

To view a topic, double-click on the topic (or press the Up or Down Arrow key to highlight the topic, and then press the Enter key).

How to Obtain Additional Information

After you select a topic, the information for that topic appears in a window. Highlighted words or phrases indicate that additional information is available. You will notice that certain words and phrases are highlighted in green letters, or in white letters on a black background. These are called hypertext terms. If you are using a mouse, double-click on the highlighted word. If you are using a keyboard, press the Tab key to move to the highlighted word, and then press the Enter key. Additional information then appears in a window.

How to Use Action Bar Choices

Several choices are available for managing information presented in the *Workplace Shell Programming Reference* . There are three pull-down menus on the action bar: the **Services** menu, the **Options** menu, and the **Help** menu.

The actions that are selectable from the **Services** menu operate on the active window currently displayed on the screen. These actions include the following:

Bookmark

Allows you to set a placeholder so you can retrieve information of interest to you.

When you place a bookmark on a topic, it is added to a list of bookmarks you have previously set. You can view the list, and you can remove one or all bookmarks from the list. If you have not set any bookmarks, the list is empty.

To set a bookmark, do the following:

1. Select a topic from the Contents.
2. When that topic appears, choose the **Bookmark** option from the **Services** pull-down.

3. If you want to change the name used for the bookmark, type the new name in the field.

4. Click on the **Place** radio button (or press the Up or Down Arrow key to select it).

5. Click on **OK** (or select it and press Enter). The bookmark is then added to the bookmark list.

Search

Allows you to find occurrences of a word or phrase in the current topic, selected topics, or all topics.

You can specify a word or phrase to be searched. You can also limit the search to a set of topics by first marking the topics in the Contents list.

To search for a word or phrase in all topics, do the following:

1. Choose the **Search** option from the **Services** pull-down.

2. Type the word or words for which to search.

3. Click on **All sections** (or press the Up or Down Arrow keys to select it).

4. Click on **Search** (or select it and press Enter) to begin the search.

5. The list of topics where the word or phrase appears is displayed.

Print

Allows you to print one or more topics. You can also print a set of topics by first marking the topics in the Contents list.

To print the document Contents list, do the following:

1. Choose **Print** from the **Services** pull-down.

2. Click on **Contents** (or press the Up or Down Arrow key to select it).

3. Click on **Print** (or select it and press Enter).

4. The Contents list is printed on your printer.

Copy

Allows you to copy a topic that you are viewing to the System Clipboard or to a file that you can edit. You will find this particularly useful for copying syntax definitions and program samples into the application that you are developing.

You can copy a topic that you are viewing in two ways:

- **Copy** copies the topic that you are viewing into the System Clipboard. If you are using a Presentation Manager* editor (for example, the System Editor) that copies or cuts (or both) to the System Clipboard, and pastes to the System Clipboard, you can easily add the copied information to your program source module.

- **Copy to file** copies the topic that you are viewing into a temporary file named TEXT.TMP. You can later edit that file by using any editor. You will find TEXT.TMP in the directory where your viewable document resides.

 To copy a topic, do the following:

 1. Expand the Contents list and select a topic.

 2. When the topic appears, choose **Copy to file** from the **Services** pull-down.

 3. The system puts the text pertaining to that topic into the temporary file named TEXT.TMP.

For information on one of the other choices in the **Services** pull-down, highlight the choice and press the F1 key.

The actions that are selectable from the **Options** menu allow you to change the way your Contents list is displayed. To expand the Contents and show all levels for all topics, choose **Expand all** from the **Options** pull-down. You can also press the Ctrl and * keys together. For information on one of the other choices in the **Options** pull-down, highlight the choice and press the F1 key.

The actions that are selectable from the **Help** menu allow you to select different types of help information. You can also press the F1 key for help information about the Information Presentation Facility (IPF).

How to Use the Programming Information

This document consists of guide and reference information that provides a detailed description of each function, message, constant, and data type. It provides language-dependent information about the functions which enable the user to generate call statements in the C Language.

Workplace Shell programming information is presented by component, such as Workplace Classes, Instance Methods, and Class Methods, for example:

```
                Contents


        + Workplace Classes
        + Instance Methods
        + Class Methods
```

By clicking on the plus sign beside "Instance Methods", you see an alphabetic list of the Workplace Shell instance methods. Selecting an instance method takes you directly into the reference information for that method.

Units of reference information are presented in selectable multiple windows or viewports. A viewport is a Presentation Manager window that can be sized, moved, minimized, maximized, or closed. By selecting a unit (in this case, an entry on the Contents list), you will see two windows displayed:

```
     Unit Title                Selection Title

     Select an item

     Syntax
     Returns
     Notes
     Related Methods
     Override
     Glossary
```

The window on the left is the primary window. It contains a list of items that are always available to you. The window on the right is the secondary window. It contains a "snapshot" of the unit information. For reference units (that is, function descriptions), this window contains the Function Syntax.

All of the information needed to understand a reference unit (or topic) is readily available to you through the primary window. The information is divided into discrete information groups, and only the appropriate information group appears for the topic that you are viewing.

The information groups for a reference unit (that is, a function description) can include all or some of the following:

- Syntax
- Parameters
- Returns
- Notes
- Related Methods
- Usage
- Override
- Glossary

This list may vary. Some topics may be omitted when they do not apply.

Information groups are displayed in separate viewports that are stacked in a third window location that overlaps the secondary window. By selecting an item (information group) in the primary window, the item is displayed in the third window location, as follows:

```
   Unit Title         Selection     Glossary

   Select an item                Select a starting
                                 letter of
      .                          glossary terms
      .
      .                          A    N
      .                          B    O
      .                          C    P
   Glossary                      .    .
                                 .    .
                                 .    .
                                 M    Z
```

By selecting successive items from the primary window, additional windows are displayed on top of the previous windows displayed in the third window location. For example, in a function description, Parameters and Return Values are items listed in the primary window. When selected, they appear one on top of the other in the third window location. Because of this, you may move the first selected (topmost) window to the left before selecting the next item. This allows simultaneous display of two related pieces of information from the "stack" of windows in the third window location, as follows:

```
Unit Title        Parameters     Return Values

Select an item
   .
   .
   .
Returns
Errors
   .
   .
   .
```

Each window can be individually closed from its system menu. All windows are closed when you close the primary window.

Some secondary windows may have the appearance of a split screen. For example, an illustration may appear in the left half of the window, and scrollable, explanatory information may appear in the right half of the window. Because illustrations may not necessarily fit into the small window size on your screen, you may maximize the secondary window for better readability.

Conventions Used in this Reference

The purpose of this reference is to give information about classes, methods, constants, and data types. It provides information about the methods which enables the user to call functions in the C programming language.

The following information is provided:

- The syntax and parameters for each method.
- The syntax of each data type and structure.

Notation Conventions

The following notation conventions are used in this reference:

NULL	The term NULL applied to a parameter is used to indicate the presence of the pointer parameter, but with no value.
NULLHANDLE	The term NULLHANDLE applied to a parameter is used to indicate the presence of the handle parameter, but with no value.
Implicit Pointer	If no entry for a data type "Pxxxxxxx" is found in Data Types, then it is implicitly a pointer to the data type "xxxxxxx". See Implicit Pointer Data Types for more information about implicit pointers.
CONSTANT NAMES	All constants are written in uppercase to match the header files. Where applicable, constant names have a prefix derived from the name of a function, message, or idea associated with the constant. For example: WM_CREATE — Window message SV_CXICON — System value CF_TEXT — Clipboard format. In this book, references to a complete set of constants with a given prefix is written as shown in the following examples: Window message — WM_* System value — SV_*
Parameters and Fields	Function parameters and data structure fields are shown in italics.

Conventions Used in Method Descriptions

The documentation of each function contains these sections:

Syntax
The method syntax describes the C-language calling syntax of the function and gives a brief description.

Programming Note
The methods in this book are spelled in mixed-case for readability but are known to the system as uppercase character strings. For example, the function "WinCreateObject" is actually the external name "WINCREATEOBJECT".

If you are using a compiler that generates a mixed-case external name, you should code the functions in uppercase.

Parameters
Each parameter is listed with its C-language data type, parameter type, and a brief description.

- All data types are written in uppercase to match the header files. A data type of "Pxxxxxxx" implicitly defines a pointer to the data type "xxxxxxx".

 The term NULL applied to a parameter indicates the presence of the parameter, with no value.

 Refer to Data Types for a complete list of all data types and their descriptions.

- There are three parameter types:

Input	Specified by the programmer.
Output	Returned by the method.
Input/Output	Specified by the programmer and modified by the method.

- A brief description is provided with each parameter. Where appropriate, restrictions are also included. In some cases, the parameter points to a structure.

Returns
A list of possible return codes is included in this section.

Remarks
This section contains additional information about the method, when required.

Related Methods
This list shows the methods (if any) that are related to the method being described.

Usage
This section describes any restrictions for using the method.

Override
This section describes when the method can be overridden.

Error Severities

Each of the error conditions given in the list of errors for each function falls into one of these areas:

Warning
The function detected a problem, but took some remedial action that enabled the function to complete successfully. The return code in this case indicates that the function completed successfully.

Error
The function detected a problem for which it could not take any sensible remedial action. The system has recovered from the problem, and the state of the system, with respect to the application, remains the same as at the time when the function was requested. The system has not even partially executed the function (other than reporting the error).

Severe Error
The function detected a problem from which the system could not reestablish its state, with respect to the application, at the time when that function was requested; that is, the system partially executed the function. This, therefore, necessitates the application performing some corrective activity to restore the system to some known state.

Unrecoverable Error

The function detected some problem from which the system could not reestablish its state, with respect to the application, at the time when that call was issued. It is possible that the application cannot perform some corrective action to restore the system to some known state.

Header Files

All functions require an "#include" statement for the system header file OS2.H:

```
#include  <OS2.H>
```

Most functions also require a "#define" statement to select an appropriate (conditional) section of the header file, and hence, the required prototype. Where this is necessary, it is shown at the head of the function definition in the form:

```
#define   INCL_name
```

Note: These "#define" statements must precede the "#include <OS2.H>" statement.

Addressing Elements in Arrays

Constants defining array elements are given values that are zero-based in C; that is, the numbering of the array elements starts at zero, not one.

For example, in the DevQueryCaps function, the sixth element of the *alArray* parameter is CAPS_HEIGHT, which is equated to 5.

Count parameters related to such arrays always mean the actual number of elements available; therefore, again using the DevQueryCaps function as an example, if all elements up to and including CAPS_HEIGHT are provided for, *lCount* could be set to (CAPS_HEIGHT+1).

In functions for which the starting array element can be specified, this is always zero-based, and so the C element number constants can be used directly. For example, to start with the CAPS_HEIGHT element, the *lStart* parameter can be set to CAPS_HEIGHT.

Implicit Pointer Data Types

A data type name beginning with "P" (for example, PERRORCODE) is likely to be a pointer to another data type (in this instance, ERRORCODE).

In the data type summary, Data Types, no explicit "typedefs" are shown for pointers; therefore, if no data type definition can be found in the summary for a data type name "Pxxxxxx", it represents a pointer to the data type "xxxxxx", for which a definition should be found in the reference.

The implicit type definition needed for such a pointer "Pxxxxxx" is:

```
typedef xxxxxx *Pxxxxxx;
```

Such definitions are provided in the header files.

Storage Mapping of Data Types

The storage mapping of the data types is dependent on the machine architecture. To be portable, applications must access the data types using the definitions supplied for the environment in which they will execute.

Double-Byte Character Set (DBCS)

Throughout this publication, you will see references to specific value for character strings. The values are for single-byte character set (SBCS). If you use the double-byte character set (DBCS), note that one DBCS character equals two SBCS characters.

Message Queues

Usually when an application thread uses a PM function, a message queue must be available for that thread. This means that before calling the function, WinCreateMsgQueue must be called by the same thread.

It is recommended that you create a message queue for every thread that calls any method, since a PM function might be used by the methods you are calling.

Programming Considerations

This section provides information you need to consider before you begin programming with Workplace Shell methods.

Method Considerations

When including the "OBJECTID=<...>" keyname/value pair in a setup string, it must be specified as the last entry in the string.

Stack Size

Existing 16-bit applications (small and tiny models) must have a 4KB stack available when they enter system calls; otherwise, the stack can overflow into the data area.

C++ Considerations

This section contains several topics you should take into consideration if you are using C++ to develop applications.

A Few Words on Typedefs

The OS/2 header files can be used with either C or C++ compilers. If __cplusplus has been defined, the header files will produce code that is compatible with C++. This is done since several of the typedefs have been changed to support C++. For example, many items that are unsigned char in the "C header files" are char when compiled with __cplusplus. For instance,

```
typedef unsigned char BYTE;
```

has changed to

```
typedef char BYTE;
```

The existing samples that are included in the IBM Developer's Toolkit for OS/2 Warp Version 3 can be compiled with either C or C++.

Note: Prior versions of the IBM Developer's Toolkit for OS/2 Warp Version 3 provided two sets of header files: one set for use with C compilers, another set for use with C++ compilers. The same set of header files are now used by either compiler.

PCSZ Data Type

If a function takes as a parameter a string that is not changed by the function, the string parameter can be declared as a "const" string, or a PCSZ. PCSZ is defined in the header files as a "const" pointer to a NULL-delimited string.

The "const" means that the function will not change the contents of the string. The use of the "const" keyword is not specific to C++, many C compilers support it as well. However, the use of PCSZ allows for better optimization by the compiler and is more semantically compatible with C++.

The PCSZ data type is defined in addition to the PSZ data type. Existing code that calls functions that use PSZ will continue to work correctly, though they may cause warning or error messages when compiled.

LINK386

The C++ compiler will provide a dynamic link library which is be used by LINK386, or other OS/2 linker, when generating error messages. This DLL will convert a compiler generated mangled name into the function prototype. If the DLL is not present, an error message will be displayed and LINK386 will display the compiler-generated mangled name in error messages.

Workplace Object Classes

This chapter contains an alphabetic listing of the Workplace object classes.

These sections contain technical reference information. For information on the System Object Model (SOM), see the *System Object Model Guide and Reference* .

Object Classes

Workplace objects are icons representative of physical objects which users deal with in the Workplace. A Workplace object is implemented as an instance of a Workplace object class. Workplace object classes are System Object Model (SOM) object classes which are descendants of the predefined Workplace object class, WPObject. For more information about SOM object classes, see *System Object Model Guide and Reference* .

All Workplace objects are descendants of a Workplace object storage class. Workplace storage object classes are responsible primarily for storing and retrieving object class data for instances of objects which are descendants (subclasses) of that storage class.

The three predefined Workplace object storage classes are:

WPAbstract	Storage class for objects stored in the user profile (OS2.INI)
WPFileSystem	Storage class for objects stored as files in the file system
WPTransient	Storage class for non-persistent objects.

Workplace Object Class Hierarchy

The following figure lists the predefined Workplace object classes in a hierarchical order. Each branch in the tree represents an immediate descendant (subclass) of a Workplace object class. The predefined SOM object class, SOMObject, is the root class for all SOM object

classes, including all Workplace object classes.

CLASS NAME	CLASS DEFINITION FILE
SOMObject	somobj.idl
SOMClass	somcls.idl
SOMClassMgr	somcm.idl
WPClassManager	wpclsmgr.idl
WPObject	wpobject.idl
WPAbstract	wpabs.idl
WPClock	wpclkm.idl
WPCountry	wpctry.idl
WPDisk	wpdisk.idl
WPLaunchPad	wplnchpd.idl
WPKeyboard	wpkeybd.idl
WPMouse	wpmouse.idl
WPPalette	wppalet.idl
WPColorPalette	wpclrpal.idl
WPFontPalette	wpfntpal.idl
WPSchemePalette	wpscheme.idl
WPPower	wppower.idl
WPPrinter	wpprint.idl
WPRPrinter	wprprint.idl
WPProgram	wppgm.idl
WPShadow	wpshadow.idl
WPNetLink	wpnetlnk.idl
WPShredder	wpshred.idl
WPSound	wpsound.idl
WPSpecialNeeds	wpspneed.idl
WPSpool	wpspool.idl
WPSystem	wpsystem.idl
WPWinConfig	wincfg.idl
WPFileSystem	wpfsys.idl
WPDataFile	wpdataf.idl
WPHtml	wphtml.idl
WPIcon	wpicon.idl
WPImageFile	wpimage.idl
WPBitmap	wpbitmap.idl
WPMet	wpmet.idl
WPPif	wppif.idl
WPPointer	wpptr.idl
WPProgramFile	wppgmf.idl
WPCommandFile	wpcmdf.idl
WPUrl	wpurl.idl
WPFolder	wpfolder.idl
WPDesktop	wpdesk.idl
WPDrives	wpdrives.idl
WPHost	wphost.idl
WPHwManager	wphwmgr.idl
WPMinWinViewer	wpmwv.idl
WPNetgrp	wpnetgrp.idl
WPNetwork	wpnetwrk.idl
WPRootFolder	wprootf.idl
WPServer	wpserver.idl
WPSharedDir	wpshdir.idl
WPStartup	wpstart.idl
WPTemplates	wptemps.idl
WPUrlFolder	wpurlfdr.idl
WPTransient	wptrans.idl
WPJob	wpjob.idl
WPDevice	wpdevice.idl
WPDevAudio	wpaudio.idl
WPDevBus	wpbus.idl
WPDevCDRom	wpcdrom.idl
WPDevCPU	wpcpu.idl
WPDevDiskette	wpdskett.idl
WPDevDisplay	wpdisply.idl
WPDevHarddrive	wphrddrv.idl
WPDevKeyboard	wpkeybdd.idl
WPDevMemory	wpmem.idl
WPDevMouse	wpmoused.idl
WPDevParallel	wpparal.idl
WPDevPeriph	wpperiph.idl
WPDevSerial	wpserial.idl
WPDevTape	wptape.idl
WPDevTimer	wptimer.idl
WPPort	wpport.idl
WPPdr	wppdr.idl
WPQdr	wpqdr.idl

Instances of some Workplace object classes cannot be created as a Workplace object. These classes are provided as base classes which provide support for descendant classes that can have instances created. Other classes are SOM classes and are described in better detail in the *System Object Model Guide and Reference* .

These classes include:

SOMObject	This is the SOM root class. All SOM classes must be descended from SOMObject. A Workplace object of this class cannot be created.
SOMClass	This is the SOM metaclass, that is, the instances of this class are class objects. A Workplace object of this class cannot be created.
SOMClassMgr	This is the SOM class manager class. A Workplace object of this class cannot be created.

WPAbstract

Class definition file: wpabs.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract

Description

This is the abstract object storage class. The storage medium for objects that are descendants of the WPAbstract class is the INI file. In other words, any object class derived from WPAbstract will have persistent storage for its instance variables in the INI file. Note that an abstract object does not have a file name, just a numeric handle that can be used to identify it. It can optionally be assigned an object id to uniquely identify it. An instance of this class can be created as a Workplace object. No instances of this class are initially created by the system.

Instance methods

There are currently no methods defined for the WPAbstract Workplace object class.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPAbstract class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpCopyObject
- wpDragOver
- wpIdentify
- wpMoveObject
- wpQueryHandle
- wpQueryIcon
- wpQueryIconData
- wpQueryNameClashOptions
- wpReplaceObject
- wpRestoreState
- wpSaveImmediate
- wpSaveState
- wpSetIconData
- wpSetTitle

Class methods

The following list shows the WPAbstract class methods.

- wpclsQuerySetting
- wpclsSetSetting

WPObject class methods

The following shows the class methods overridden by the WPAbstract class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsObjectFromHandle
- wpclsQueryTitle

WPBitmap

Class definition file: wpbitmap.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPDataFile
 WPImageFile
 WPBitmap

Description

This is the WPBitmap object class. All bit maps are of this class. These bit map files are templatable. An instance of this class can be created as a Workplace object. Instances of this class are created initially by the system for each bit map in the system.

Instance methods

There are currently no methods defined for the WPBitmap Workplace object class.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPBitmap class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpFormatDragItem
- wpInitData
- wpQueryBitmapData
- wpQueryBitmapHandle
- wpQueryBitmapInfoHeader
- wpReadImageFile
- wpRefresh
- wpSetBitmapData
- wpSetupOnce
- wpUnInitData
- wpWriteImageFile

Class methods

There are currently no class methods defined for the WPBitmap Workplace object class.

WPObject class methods

The following list shows all the class methods overridden by the WPBitmap class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryDefaultHelp
- wpclsQueryStyle
- wpclsQueryTitle

WPFileSystem class methods

The following list shows all the class methods overridden by the WPBitmap class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryInstanceFilter
- wpclsQueryInstanceType

WPClassManager

Class definition file: wpabs.idl

Class hierarchy
SOMObject
 SOMClassMgr
 WPClassManager

Description

This is the Workplace class manager object class. It is responsible for modifying SOMClassMgr to support object class replacement, and to utilize the Workplace class registration mechanism so that a Workplace class can be found using somFindClass. See the online book *System Object Model Programming Reference* for more information on SOMClassMgr.

Instance methods

The following list shows all the WPClassManager instance methods:

- wpGetTrueClassName
- wpModuleForClass
- wpReplacementIsInEffect

SOMObject instance methods

The following list shows all the methods overridden by the SOMObject class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct
- somInit

SOMClassMgr instance methods

The following list shows all the methods overridden by the SOMClassMgr class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- somFindClass
- somFindClsInFile
- somLocateClassFile
- somMergeInto
- somRegisterClass
- somUnregisterClass

WPClock

Class definition file: wpclock.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPClock

Description

This is the system clock object class. An instance of this class can be created as a Workplace object. An instance of this class is created initially by the system. It has the title "System Clock" and resides in the "System Setup" folder.

Instance methods

The following list shows all the WPClock instance methods:

- wpAddClockAlarmPage
- wpAddClockDateTimePage
- wpAddClockView1Page
- wpAddClockView2Page

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPClock class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectWindowPage
- wpAddSettingsPages
- wpFilterMenu
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyMenu
- wpOpen
- wpRestoreState
- wpSaveState

Class methods

There are currently no class methods defined for the WPClock Workplace object class.

WPObject class methods

The following list shows all the class methods overridden by the WPClock class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQueryStyle
- wpclsQueryTitle

WPColorPalette

Class definition file: wpclrpal.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPPalette
 WPColorPalette

Description

This is the color palette object class. An instance of this class can be created as a Workplace object. Two instances of this class are created initially by the system, with the titles "Mixed Color Palette" and "Solid Color Palette", both residing in the "System Setup" folder.

Instance methods

There are currently no methods defined for the WPColorPalette Workplace object class.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPColorPalette class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpInitData
- wpMenuItemHelpSelected
- wpSetup

wpSetup override by WPColorPalette

The following table shows the keyname-value pairs added by the WPColorPalette class.

KEYNAME	VALUE	DESCRIPTION
AUTOSETUP	HIRES	This sets the number of default colors in the color palette to the 256-color Mixed Color Palette.
	LORES	This sets the number of default colors in the color palette to the 16-color Solid Color Palette.
COLORS	RGB values	These are the initial color values of each cell in the color palette. The values for each cell are separated by commas. This is equivalent to calling the wpSetupCell method. The RGB value must be presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue, values ranging between 0x00 and 0xFF (0-255).
XCELLCOUNT	columns	Number of X cells as decimal digits. For Solid Color Palette, AUTOSETUP=LORES must be specified, and the default value is 8. For Mixed Color Palette, AUTOSETUP=HIRES must be specified, and the default value is 16.
YCELLCOUNT	rows	Number of Y cells as decimal digits. For Solid Color Palette, AUTOSETUP=LORES must be specified, and the default value is 2. For Mixed Color Palette, AUTOSETUP=HIRES must be specified, and the default value is 16.

Example

```
pszSetupString="COLORS=0x000000,"
                      "0xFFFFFF,"
                      "0xFF23B4,"
                      "0x505050,"
                      "0x00FFFF,"
                      "0xA0A0A0";
```

WPPalette methods

The following list shows all the methods overridden by the WPColorPalette class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpDragCell
- wpEditCell
- wpPaintCell
- wpQueryPaletteHelp

- wpSelectCell

Class methods

There are currently no class methods defined for the WPColorPalette Workplace object class.

WPObject class methods

The following list shows all the class methods overridden by the WPColorPalette class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsCreateDefaultTemplates
- wpclsQueryDefaultHelp
- wpclsQueryIconData
- wpclsQueryStyle
- wpclsQueryTitle

WPPalette class methods

The following list shows all the class methods overridden by the WPColorPalette class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryEditString

WPCommandFile

Class definition file: wpcmdf.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPDataFile
 WPProgramFile
 WPCommandFile

Description

This is the command file object class. All DOS and OS/2 command/batch files are of this class. These program files are templatable. An instance of this class can be created as a Workplace object. Instances of this class are created initially by the system for each command file in the system. Each instance has a title corresponding to the file name of the command file it represents and resides in a folder corresponding to the physical directory in which the program file resides.

Instance methods

There are currently no methods defined for the WPCommandFile Workplace object class.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPCommandFile class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpSetupOnce
- wpPrintObject

Class methods

There are currently no class methods defined for the WPCommandFile Workplace object class.

WPObject class methods

The following list shows all the class methods overridden by the WPCommandFile class. These methods are overridden to modify the

behavior defined by an ancestor class.

- wpclsQueryStyle
- wpclsQueryTitle

WPFileSystem class methods

The following list shows all the class methods overridden by the WPCommandFile class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryInstanceFilter
- wpclsQueryInstanceType

WPCountry

Class definition file: wpctry.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPCountry

Description

This is the country object class. An instance of this class can be created as a Workplace object. An instance of this class is created initially by the system. It has the title, "Country" and resides in the "System Setup" folder.

Instance methods

The following list shows all the WPCountry instance methods:

- wpAddCountryDatePage
- wpAddCountryNumbersPage
- wpAddCountryPage
- wpAddCountryTimePage

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPCountry class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectWindowPage
- wpAddSettingsPages
- wpFilterMenu
- wpInitData

Class methods

There are currently no class methods defined for the WPCountry Workplace object class.

WPObject class methods

The following list shows all the class methods overridden by the WPCountry class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQueryStyle
- wpclsQueryTitle

WPDataFile

Class definition file: wpdataf.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPDataFile

Description

This is the data file object class. An instance of this class can be created as a Workplace object and is created initially by the system in its template form. The template form of this class has the title "Data file" and resides in the "Templates" folder.

Menu ID list

The following list shows the WPMENUID menu IDs for this class:

Menu item text	Menu ID
Open	WPMENUID_OPEN

Instance methods

The following list shows all the WPDataFile methods:

- wpAddBecomePage
- wpAddFileTypePage
- wpPrintMetaFile
- wpPrintPifFile
- wpPrintPrinterSpecificFile
- wpPrintUnknownFile
- wpQueryAssociatedFileIcon
- wpQueryAssociatedProgram
- wpQueryHandleFromContents
- wpQueryPointerFromContents
- wpRenderFromClipboard
- wpSetAssociatedFileIcon
- wpSetContentsFromHandle
- wpSetContentsFromPointer

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPDataFile class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectWindowPage
- wpAddSettingsPages
- wpConfirmKeepAssoc
- wpConfirmRenameFileWithExt
- wpDragOver
- wpFormatDragItem
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyMenu
- wpOpen
- wpPrintObject
- wpQueryDefaultHelp
- wpQueryIcon
- wpRender
- wpRestoreState
- wpSetDefaultView
- wpSetRealName

WPFileSystem instance methods

The following list shows all the methods overridden by the WPDataFile class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpSetType

Class methods

The following list shows all the WPDataFile class methods:

- wpclsAddClipboardAssoc
- wpclsEnumClipboardClasses

WPObject class methods

The following list shows all the class methods overridden by the WPDataFile class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQuerySettingsPageSize
- wpclsQueryStyle
- wpclsQueryTitle

WPDesktop

Class definition file: wpdesk.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPFolder
 WPDesktop

Description

This is the Workplace desktop object class. An instance of this class can be created as a Workplace object. An instance of this class is created initially by the system. It has the title "Desktop" and initially resides in the root directory of the drive containing the user profile.

Menu ID list

The following list shows the WPMENUID menu IDs for this class:

Menu item text	Menu ID
Lockup now	WPMENUID_LOCKUP
Logoff now	WPMENUID_LOGOFF This menu id is not available on OS/2 Warp version 3 or lower.
Shut down	WPMENUID_SHUTDOWN
System setup	WPMENUID_SYSTEMSETUP

Instance methods

The following list shows all the WPDesktop methods:

- wpAddDesktopArcRest1Page
- wpAddDesktopDefDT1Page
- wpAddDesktopLockup1Page
- wpAddDesktopLockup2Page
- wpAddDesktopLockup3Page
- wpIsCurrentDesktop
- wpQueryAutoLockup
- wpQueryDefaultDropOp
- wpQueryLockupAutoDim

- wpQueryLockupBackground
- wpQueryLockupFullScreen
- wpQueryLockupOnStartup
- wpQueryLockupTimeout
- wpSetAutoLockup
- wpSetDefaultDropOp
- wpSetLockupAutoDim
- wpSetLockupBackground
- wpSetLockupFullScreen
- wpSetLockupOnStartup
- wpSetLockupTimeout

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPDesktop class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddSettingsPages
- wpFilterMenu
- wpInitData
- wpMenuItemSelected
- wpModifyMenu
- wpRestoreState
- wpSaveState
- wpSetup

wpSetup override by WPDesktop

The following table shows the keyname-value pairs added by the WPDesktop class.

KEYNAME	VALUE	DESCRIPTION
AUTOLOCKUP	YES	This sets the auto-lockup feature, so that the keyboard and mouse will automatically lock up after the specified number of minutes of inactivity.
	NO	This turns off the autolockup feature, so that the system will not automatically lock the keyboard and mouse, no matter how long they are inactive.
LOCKUPAUTODIM	YES	This sets the autodim feature, so that the screen blanks out and a floating lock icon is displayed 2 minutes after the keyboard and mouse are locked.
	NO	This turns off the autodim feature.
LOCKUPBACKGROUND	N,M,S,B,C	This sets the lockup background, where: N = Image file name. This name must be the fully-qualified path of the image file. "(none)" or "?:\" is permitted to indicate the boot drive. M = Image mode. This mode can be one of the following: N = Normal image T = Titled image S = Scaled image S = Scaling factor.

		B = Background type. This can be one of the following: I = Image C = Color only C = Background color. This color can be 3 numbers representing RGB values or the default, in the following format: {R-value G-value B-value} For example: "BACKGROUND= C:\OS2\BITMAP\OS2LOGO.BMP,S, 3,I" or "BACKGROUND= (none),,,C,0 128 128"
LOCKUPFULLSCREEN	YES	This specifies that the entire screen is taken up by the lockup background image. The lockup background specified by the LOCKUPBACKGROUND keyname is displayed when the system locks the keyboard and mouse.
	NO	This specifies that when the system locks up, a message box is to be displayed prompting you to enter your lockup password.
LOCKUPONSTARTUP	YES	This specifies that the keyboard and mouse are automatically locked when the system is started or restarted.
	NO	This specifies that the keyboard and mouse are not automatically locked when the system is started or restarted.
LOCKUPTIMEOUT	n	This specifies the number of minutes of keyboard and mouse inactivity that will cause the system to automatically lock the keyboard and mouse.

Class methods

The following list shows the WPDesktop class methods.

- wpclsQueryActiveDesktop
- wpclsQueryActiveDesktopHWND
- wpclsQueryObjectFromFrame

WPObject class methods

The following list shows all the class methods overridden by the WPDesktop class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQueryIconDataN
- wpclsQuerySettingsPageSize
- wpclsQueryStyle
- wpclsQueryTitle

WPDevAudio

Class definition file: wpaudio.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient
 WPDevice
 WPDevAudio

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This object class represents audio hardware adapters or devices on a computer. An instance of this class can be created as a Workplace object.

Instance methods

There are currently no instance methods defined for the WPDevAudio Workplace object class.

WPObject instance methods

There are currently no WPObject instance methods overridden for the WPDevAudio Workplace object class.

Class methods

There are currently no class methods defined for the WPDevAudio Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPDevAudio class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryIconData
- wpclsQueryTitle

WPDevBus

Class definition file: wpbus.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient
 WPDevice
 WPDevBus

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This object class represents a hardware bus on a computer. An instance of this class can be created as a Workplace object.

Instance methods

There are currently no instance methods defined for the WPDevBus Workplace object class.

WPObject instance methods

There are currently no WPObject instance methods overridden for the WPDevBus Workplace object class.

Class methods

There are currently no class methods defined for the WPDevBus Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPDevBus class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryIconData
- wpclsQueryTitle
- wpclsUnInitData

WPDevCDRom

Class definition file: wpcdrom.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient
 WPDevice
 WPDevCDRom

Description This class is specific to Version 4, or higher, of the OS/2 operating system.

This object class represents CDRom hardware on a computer. An instance of this class can be created as a Workplace object.

Instance methods

There are currently no instance methods defined for the WPDevCDRom Workplace object class.

WPObject instance methods

There are currently no WPObject instance methods defined for the WPDevCDRom Workplace object class.

Class methods

There are currently no class methods defined for the WPDevCDRom Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPDevCDRom class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryIconData
- wpclsQueryTitle
- wpclsUnInitData

WPDevCPU

Class definition file: wpcpu.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient
 WPDevice
 WPDevCPU

Description This class is specific to Version 4, or higher, of the OS/2 operating system.

This object class represents a CPU on a computer. An instance of this class can be created as a Workplace object.

Instance methods

There are currently no instance methods defined for the WPDevCPU Workplace Object class.

WPObject instance methods

There are currently no WPObject instance methods overridden for the WPDevCPU Workplace object class.

Class methods

There are currently no class methods defined for the WPDevCPU Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPDevCPU class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryIconData
- wpclsQueryTitle
- wpclsUnInitData

WPDevDiskette

Class definition file: wpdskett.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient
 WPDevice
 WPDevDiskette

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This object class represents a diskette adapter or device on a computer. An instance of this class can be created as a Workplace object.

Instance methods

There are currently no instance methods defined for the WPDevDiskette Workplace Object class.

WPObject instance methods

There are currently no WPObject instance methods overridden by the WPDevDiskette class.

Class methods

There are currently no class methods defined for the WPDevDiskette Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPDevDiskette class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryIconData
- wpclsQueryTitle
- wpclsUnInitData

WPDevDisplay

Class definition file: wpdisply.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient

WPDevice
WPDevDisplay

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This object class represents the video display hardware on a computer. An instance of this class can be created as a Workplace object.

Instance methods

There are currently no instance methods defined for the WPDevDisplay Workplace object class.

WPObject instance methods

There are currently no WPObject instance methods overridden by the WPDevDisplay class.

Class methods

There are currently no class methods defined for the WPDevDisplay Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPDevDisplay class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryIconData
- wpclsQueryTitle
- wpclsUnInitData

WPDevHarddrive

Class definition file: wphrddrv.idl

Class hierarchy
SOMObject
WPObject
WPTransient
WPDevice
WPDevHarddrive

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This object class represents a harddrive adapter or device on a computer. An instance of this class can be created as a Workplace object.

Instance methods

There are currently no instance methods defined for the WPDevHarddrive Workplace Object class.

WPObject instance methods

There are currently no WPObject instance methods overridden by the WPDevHarddrive class.

Class methods

There are currently no class methods defined for the WPDevHarddrive Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPDevHarddrive class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryIconData
- wpclsQueryTitle
- wpclsUnInitData

WPDevice

Class definition file: wpdevice.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient
 WPDevice

Description This class is specific to Version 4, or higher, of the OS/2 operating system.

This object class represents hardware adapters and devices on a computer. An instance of this class can be created as a Workplace object.

Instance methods

The following list shows all the WPDevice methods:

- wpAddDeviceDevice1Page
- wpAddDeviceDevice2Page
- wpAddDeviceDMA1Page
- wpAddDeviceIO1Page
- wpAddDeviceIRQ1Page
- wpAddDeviceMemory1Page

WPObject instance methods

The following list shows all the methods overridden by the WPDevice class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddSettingsPages
- wpCnrInsertObject
- wpInitData
- wpModifyPopupMenu
- wpQueryDetailsData
- wpRestoreState
- wpSaveState
- wpSetup
- wpUnInitData

Class methods

There are currently no class methods defined for the WPDevice Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPDevice class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryIconData
- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryDetailsInfo
- wpclsQueryStyle
- wpclsQueryTitle
- wpclsUnInitData

WPDevKeyboard

Class definition file: wpkeybdd.idl

Class hierarchy
SOMObject

WPObject
WPTransient
WPDevice
WPDevKeyboard

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This object class represents a keyboard adapter or device on a computer. An instance of this class can be created as a Workplace object.

Instance methods

There are currently no instance methods defined for the WPDevKeyboard Workplace Object class.

WPObject instance methods

There are currently no WPObject instance methods overridden by the WPDevKeyboard class.

Class methods

There are currently no class methods defined for the WPDevKeyboard Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPDevKeyboard class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryIconData
- wpclsQueryTitle
- wpclsUnInitData

WPDevMemory

Class definition file: wpmem.idl

Class hierarchy
SOMObject
WPObject
WPTransient
WPDevice
WPDevMemory

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This object class represents a memory adapter or device on a computer. An instance of this class can be created as a Workplace object.

Instance methods

There are currently no instance methods defined for the WPDevMemory Workplace object class.

WPObject instance methods

There are currently no WPObject instance methods overridden by the WPDevMemory class.

Class methods

There are currently no class methods defined for the WPDevMemory Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPDevMemory class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryIconData
- wpclsQueryTitle

- wpclsUnInitData

WPDevMouse

Class definition file: wpmoused.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient
 WPDevice
 WPDevMouse

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This object class represents a Mouse adapter or device on a computer. An instance of this class can be created as a Workplace object.

Instance methods

There are currently no instance methods defined for the WPDevMouse Workplace object class.

WPObject instance methods

There are currently no WPObject instance methods overridden by the WPDevMouse class.

Class methods

There are currently no class methods defined for the WPDevMouse Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPDevMouse class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryIconData
- wpclsQueryTitle
- wpclsUnInitData

WPDevParallel

Class definition file: wpparal.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient
 WPDevice
 WPDevParallel

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This object class represents a Parallel port adapter or device on a computer. An instance of this class can be created as a Workplace object.

Instance methods

There are currently no instance methods defined for the WPDevParallel Workplace object class.

WPObject instance methods

There are currently no WPObject instance methods overridden by the WPDevParallel class.

Class methods

There are currently no class methods defined for the WPDevParallel Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPDevParallel class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryIconData
- wpclsQueryTitle
- wpclsUnInitData

WPDevPeriph

Class definition file: wpperiph.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient
 WPDevice
 WPDevPeriph

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This object class represents a Peripheral adapter or device on a computer. An instance of this class can be created as a Workplace object.

Instance methods

There are currently no instance methods defined for the WPDevPeriph Workplace object class.

WPObject instance methods

There are currently no WPObject instance methods overridden by the WPDevPeriph class.

Class methods

There are currently no class methods defined for the WPDevPeriph Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPDevPeriph class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryIconData
- wpclsQueryTitle
- wpclsUnInitData

WPDevSerial

Class definition file: wpserial.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient
 WPDevice

WPDevSerial

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This object class represents a Serial port adapter or device on a computer. An instance of this class can be created as a Workplace object.

Instance methods

There are currently no instance methods defined for the WPDevSerial Workplace object class.

WPObject instance methods

There are currently no WPObject instance methods overridden by the WPDevSerial class.

Class methods

There are currently no class methods defined for the WPDevSerial Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPDevSerial class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryIconData
- wpclsQueryTitle
- wpclsUnInitData

WPDevTape

Class definition file: wptape.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient
 WPDevice
 WPDevTape

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This object class represents a Tape backup adapter or device on a computer. An instance of this class can be created as a Workplace object.

Instance methods

There are currently no instance methods defined for the WPDevTape Workplace object class.

WPObject instance methods

There are currently no WPObject instance methods overridden by the WPDevTape class.

Class methods

There are currently no class methods defined for the WPDevTape Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPDevTape class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryIconData
- wpclsQueryTitle
- wpclsUnInitData

WPDevTimer

Class definition file: wptimer.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient
 WPDevice
 WPDevTimer

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This object class represents a Timer adapter or device on a computer. An instance of this class can be created as a Workplace object.

Instance methods

There are currently no instance methods defined for the WPDevTimer Workplace object class.

WPObject instance methods

There are currently no WPObject instance methods overridden by the WPDevTimer class.

Class methods

There are currently no class methods defined for the WPDevTimer Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPDevTimer class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryIconData
- wpclsQueryTitle
- wpclsUnInitData

WPDisk

Class definition file: wpdisk.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPDisk

Description

This is the file-system device object class. It is used to represent all types of file-system devices including:

- CD ROM drives
- Fixed disk drives
- Floppy diskette drives
- Magnetic tape drives
- Network drives
- PCMCIA** devices
- Ram drives

An instance of this class can be created as a Workplace object. Shadow instances of this class are created initially by the system for each file-system device available. Each instance has a title corresponding to a file-system device and resides in the Drives folder. If a primary, local, removable, media file-system device such as a floppy drive exists, an instance of this class will also be created on the desktop, and will represent this drive. These instances actually appear as WPShadow objects which are shadows of WPDisk objects.

Menu ID list

The following list shows the WPMENUID menu IDs for this class:

Menu item text	Menu ID
Check disk	WPMENUID_CHKDSK
Copy disk	WPMENUID_COPYDSK
Create partition	WPMENUID_FIXDSK
Deselect all	WPMENUID_DESELALL
Details view	WPMENUID_DETAILS
Eject disk	WPMENUID_EJECTDISK
Find	WPMENUID_FIND
Format disk	WPMENUID_FORMAT
Icon view	WPMENUID_ICON
Lock disk	WPMENUID_LOCKDISK
Refresh now	WPMENUID_REFRESH
Select	WPMENUID_SELECT
Select all	WPMENUID_SELALL
Sort	WPMENUID_SORT
Tree view	WPMENUID_TREE
Unlock disk	WPMENUID_UNLOCKDISK

Instance methods

The following list shows the WPDisk methods:

- wpAddDiskDetailsPage
- wpEjectDisk
- wpLockDrive
- wpQueryDriveLockStatus
- wpQueryLogicalDrive
- wpQueryRootFolder
- wpSetCorrectDiskIcon

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPDisk class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectWindowPage
- wpAddSettingsPages
- wpDragOver
- wpDrop
- wpFilterMenu
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyMenu
- wpOpen
- wpQueryDefaultHelp
- wpQueryIconData
- wpRestoreState
- wpSaveState
- wpSetIconData
- wpSetTitle
- wpSetup
- wpUnInitData
- wpViewObject

wpSetup override by WPDisk

The following table shows the keyname-value pairs added by the WPDisk class.

```
KEYNAME    VALUE           DESCRIPTION

DRIVENUM   DRIVENUM=dd     Logical drive number (1-26).
```

The following list shows all the methods overridden by the WPDisk class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpMenuItemHelpSelected
- wpMenuItemSelected

Class methods

There are currently no class methods defined for the WPDisk Workplace object class.

WPObject class methods

The following list shows all the class methods overridden by the WPDisk class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQuerySettingsPageSize
- wpclsQueryStyle
- wpclsQueryTitle

WPDrives

Class definition file: wpdrives.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPFolder
 WPDrives

Description

This is the file-system device folder object class. An instance of this class can be created as a Workplace object. An instance of this class is created initially by the system. It has the title "Drives" and resides in the "OS/2 System" folder.

Menu ID list

The following list shows the WPMENUID menu IDs for this class:

Menu item text	Menu ID
Create partition	WPMENUID_FIXDSK

Instance methods

The following list shows the WPDrives methods:

- wpAddReserveDriveLetterPage

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject methods

The following list shows all the methods overridden by the WPDrives class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddSettingsPages
- wpAddToContent
- wpFilterMenu
- wpMenuItemHelpSelected
- wpMenuItemSelected

- wpModifyMenu

WPFolder methods

The following list shows all the methods overridden by the WPDrives class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpPopulate

Class methods

There are currently no class methods defined for the WPDrives Workplace object class.

WPObject class methods

The following list shows all the class methods overridden by the WPDrives class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryIconData
- wpclsQueryIconDataN
- wpclsQueryStyle
- wpclsQueryTitle

WPFileSystem

Class definition file: wpfsys.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem

Description

This is the file-system object storage class. WPFileSystem is the storage class that represents all file-system objects including directory (folder), data file, executable file, and root directory (drive) objects. This class also provides persistent storage of instance variables for all classes derived from it. Persistent data for instances of WPFileSystem subclasses are stored in the Extended Attributes (EAs) of the file or directory.

This is a base storage class; it is not intended for instances of this class to be created.

Instance methods

The following list shows all the WPFileSystem instance methods:

- wpAddFileMenuPage
- wpAddFile1Page
- wpAddFile2Page
- wpAddFile3Page
- wpAddUserItemsToPopupMenu
- wpConfirmKeepAssoc
- wpConfirmRenameFileWithExt
- wpIsDiskSwapped
- wpPrintPlainTextFile
- wpQueryAttr
- wpQueryCreation
- wpQueryDateInfo
- wpQueryDisk
- wpQueryEASize
- wpQueryEASupport
- wpQueryFilename
- wpQueryFileSize
- wpQueryLastAccess
- wpQueryLastWrite
- wpQueryRealName
- wpQueryRefreshFlags
- wpQueryType
- wpRefresh
- wpSetAttr

- wpSetDateInfo
- wpSetFileSizeInfo
- wpSetRealName
- wpSetRefreshFlags
- wpSetTitleAndRenameFile
- wpSetType
- wpVerifyUpdateAccess

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPFileSystem class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddSettingsPages
- wpAppendObject
- wpCopyObject
- wpCreateFromTemplate
- wpDoesObjectMatch
- wpDraggedOverObject
- wpDroppedOnObject
- wpFilterMenu
- wpFormatDragItem
- wpIdentify
- wpIsDeleteable
- wpMenuItemSelected
- wpModifyMenu
- wpMoveObject
- wpOpen
- wpQueryDetailsData
- wpQueryHandle
- wpQueryIcon
- wpQueryIconData
- wpQueryNameClashOptions
- wpQueryStyle
- wpReplaceObject
- wpRestoreState
- wpSaveImmediate
- wpSaveState
- wpSetIconData
- wpSetTitle
- wpSwitchTo
- wpUnInitData
- wpViewObject

Class methods

The following list shows the WPFileSystem class methods:

- wpclsFileSysExists
- wpclsQueryAwakeObject
- wpclsQueryInstanceClass
- wpclsQueryInstanceFilter
- wpclsQueryInstanceType
- wpclsQueryObjectFromPath

WPObject class methods

The following list shows all the class methods overridden by the WPFileSystem class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsObjectFromHandle
- wpclsQueryDetailsInfo
- wpclsQueryExtendedCriteria
- wpclsQuerySettingsPageSize
- wpclsNew
- wpclsQueryTitle

WPFolder

Class definition file: wpfolder.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPFolder

Description

This is the folder object class. An instance of this class can be created as a Workplace object. An instance of this class is created initially by the system in its template form. It has the title "Folder" and resides in the "Templates" folder. Every directory in the system is represented by a WPFolder object.

Menu ID list

The following list shows the WPMENUID menu IDs for this class:

Menu item text	Menu ID
Arrange	WPMENUID_ARRANGE
Deselect all	WPMENUID_DESELALL
Details view	WPMENUID_DETAILS
Find	WPMENUID_FIND
Icon view	WPMENUID_ICON
Open as	WPMENUID_OPEN
Refresh now	WPMENUID_REFRESH
Select all	WPMENUID_SELALL
Select	WPMENUID_SELECT
Sort	WPMENUID_SORT
Tree view	WPMENUID_TREE

Instance methods

The following list shows how all the WPFolder methods are related within functional areas:

Settings notebook methods

- wpAddFolderBackgroundPage
- wpAddFolderIncludePage
- wpAddFolderMenu2Page
- wpAddFolderSelfClosePage
- wpAddFolderSortPage
- wpAddFolderView1Page
- wpAddFolderView2Page
- wpAddFolderView3Page

Object position methods

- wpFreeIconPosData
- wpInitIconPosData
- wpQueryIconPosition
- wpQueryIconViewPos
- wpQueryNextIconPos
- wpSetNextIconPos
- wpStoreIconPosData

Folder attributes methods

- wpIsDetailsColumnVisible
- wpIsSortAttribAvailable
- wpModifyFldrFlags
- wpQueryFldrAttr
- wpQueryFldrBackground
- wpQueryFldrDetailsClass
- wpQueryFldrFlags
- wpQueryFldrFont
- wpQueryFldrSort
- wpQueryFldrSortClass
- wpQueryIconTextBackgroundColor

- wpQueryIconTextColor
- wpQueryIconTextVisibility
- wpQueryMenuBarVisibility
- wpQueryShadowTextColor
- wpSetDetailsColumnVisibility
- wpSetFldrAttr
- wpSetFldrBackground
- wpSetFldrDetailsClass
- wpSetFldrFlags
- wpSetFldrFont
- wpSetFldrSort
- wpSetFldrSortClass
- wpSetIconTextBackgroundColor
- wpSetIconTextColor
- wpSetIconTextVisibility
- wpSetMenuBarVisibility
- wpSetShadowTextColor
- wpSetSortAttribAvailable

Folder state methods

- wpHideFldrRunObjs
- wpQueryOpenFolders
- wpRestoreFldrRunObjs
- wpStoreFldrRunObjs

Folder notification methods

- wpFlushNotifications

Folder content methods

- wpAddFirstChild
- wpAddToContent
- wpContainsFolders
- wpDeleteContents
- wpDeleteFromContent
- wpPopulate
- wpQueryContent
- wpSearchFolder

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPFolder class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectWindowPage
- wpAddSettingsPages
- wpConfirmDelete
- wpDelete
- wpDisplayMenu
- wpDragOver
- wpDrop
- wpFilterMenu
- wpFormatDragItem
- wpFree
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyMenu
- wpMoveObject
- wpObjectReady
- wpOpen
- wpQueryDefaultHelp
- wpQueryDefaultView
- wpRender
- wpRenderComplete
- wpRestoreState

- wpSaveState
- wpSetIconData
- wpSetup
- wpUnInitData

wpSetup override by WPFolder

The following table shows the keyname-value pairs supported by the WPFolder class.

KEYNAME	VALUE	DESCRIPTION
ALWAYSSORT	YES	Sort order is always maintained. Opening and adding an object to a folder may take longer if the sort order is being maintained.
	NO	Sort order is not maintained. This is the default value.
BACKGROUND	N,M,S,B,C	This sets the folder background, where: N = Image file name. This name must be the fully-qualified path of the image file. "(none)" or "?:\" is permitted to indicate the boot drive. M = Image mode. This mode can be one of the following: N = Normal image T = Titled image S = Scaled image S = Scaling factor. B = Background type. This can be one of the following: I = Image C = Color only C = Background color. This color can be 3 numbers representing RGB values or the default, in the following format: {R-value G-value B-value} For example: "BACKGROUND= C:\OS2\BITMAP\OS2LOGO.BMP,S, 3,I" or "BACKGROUND= (none),,,C,0 128 128"
DEFAULTSORT	index	This sets the default sort attribute on the sort menu. The index value specifies the index number of the details data item to be used as the default sort property. For example, if this keyname is set to 0, the folder's popup menu sort option has the first item checked. The default is set to 0.
DEFAULTVIEW	ICON	This sets the default open view to the ICON (or CONTENTS) view.
	TREE	This sets the default open view to the TREE view.
	DETAILS	This sets the default open view to the DETAILS view. The default view of the containing folder.

DETAILSCLASS	classname	This sets the object class for which the details are displayed in details view. The default object class is WPFileSystem.
DETAILSFONT	fontname	Specifies the font to be used in details view. For example: 10.Helv The default value is 8.Helv.
DETAILSSHADOWCOLOR	color	This sets the color of the text for a shadow object in details view. The color value may be the name of a color or 3 numbers representing RBG values.
DETAILSTEXTCOLOR	color	This set the color of the text for a normal object in details view. The color value may be the name of a color or 3 numbers representing RBG values.
DETAILSTODISPLAY	index[,...]	This sets the details to be displayed for a given class. The index is the column index, starting at 0, of the field to display. For example, "DETAILSTODISPLAY=0,2" specifies that only the first and third details data items are to be displayed. The default value is to display all fields.
DETAILSVIEW	NORMAL	This specifies normal size icons.
	MINI	This specifies small icons. This is the default.
ICONFONT	fontname	Specifies the font to be used in icon view. For example: 10.Helv
ICONNFILE	index,filename	This sets the file name of the animation (open folder) icon. The "index" value must be set to 1. The specified file contains the folder's open folder icon.
ICONGRIDSIZE		This sets the icon spacing for icon view.
ICONFILE	index,filename	This sets the animation icon to be used for this folder. Filename specifies the .ICO file containing the icon. The value of index must be 1 (indicating the open folder icon).
ICONNRESOURCE	index,id,modname	This sets the resource of the animation (open folder) icon. The "index" value must be set to 1. The "id" is the identity of an icon resource in the "module" dynamic link library (DLL). The specified resource is the folder's open folder icon.
ICONSHADOWCOLOR	color	This sets the color of the text associated with the shadow icons in icon view. The color value may be the name of a color or 3 numbers representing RBG values.
ICONTEXTBACKGROUNDCOLOR	color	This sets the color of the background for all normal text displayed in icon view, tree view and details view. The color value may be the name of a color or 3 numbers representing RBG values.
ICONTEXTCOLOR	color	This sets the color of the text associated with normal icons in icon view. The color value may be the name of a color or 3 numbers

		representing RBG values.
ICONTEXTVISIBLE	YES	This sets the icon text visible property, so that the names of the objects contained within the folder are displayed in icon view.
	NO	This resets the icon text visible property, so that the names of the objects are not displayed.
ICONVIEW	s1[,s2,...sn]	This sets icon view to the specified styles. Possible values are: GRIDDED or NONGRID FLOWED or NONFLOWED MINI, NORMAL or INVISIBLE The default value is NONGRID,NORMAL
ICONVIEWPOS	x,y,cx,cy	This sets the initial icon view position and size of the folder on the screen. These values are percentages (0-100) of the screen size.
MENUBAR	YES	This sets the menu bar property, so that the menu bar is displayed in an open view of the folder.
	NO	This resets the menu bar propert, so that the menu bar is not displayed.
OPEN	ICON	This opens the icon view when object is created or when WinSetObjectData is called.
	TREE	This opens the tree view when object is created or when WinSetObjectData is called.
	DETAILS	This opens the details view when object is created or when WinSetObjectData is called. The default is to not automatically open a view.
REMOVEFONTS	YES	This removes all instance fonts from the folder.
	NO	This does not remove instance fonts from the folder. This is the default value.
SHOWALLINTREEVIEW	YES	This set the show all in tree view property, so that all objects are displayed in an open tree view of the folder.
	NO	This resets the show all in tree view property, so that only folders are displayed in an open tree view of the folder.
SORTBYATTR	index[,...]	This sets the attributes to sort by. Index specifies a list of index numbers of details data to be shown in the "Sort by Attribute" list for the folder. For example, "SORTBYATTR=0,2" specifies that only the first and third details data items are to be shown as potential sort properties.
SORTCLASS	classname	This sets the class object to sort by. The default class object is WPFileSystem.
TREEFONT	fontname	This specifies the font to be used in tree view. For example:

		10.System Proportional
TREESHADOWCOLOR	color	This sets the color of the text associated with the shadow icons in tree view. The color value may be the name of a color or 3 numbers representing RBG values.
TREETEXTCOLOR	color	This sets the color of the text associated with normal icons in tree view. The color value may be the name of a color or 3 numbers representing RBG values.
TREETEXTVISIBLE	YES	This sets the text visible in tree view property, so that the names of the objects contained within the folder are displayed in tree view.
	NO	This resets the text visible in tree view property, so that the names of the objects are not displayed.
TREEVIEW	s1[,s2,...sn]	This sets the tree view to the specified styles. Possible values are: INVISIBLE LINES MINI NOLINES NORMAL The default value is LINES, NORMAL.
view styles	FLOWED	Flowed list items.
	NONFLOWED	Non-flowed list items.
	NONGRID	Non-gridded icon view.
	NORMAL	Normal size icons.
	MINI	Small icons.
	INVISIBLE	No icons.
	LINES	Lines in tree view.
	NOLINES	No lines in tree view.
WORKAREA	YES	This sets the folder so it is a workarea.
	NO	This sets the folder so it is not a workarea. This is the default value.

WPFileSystem instance methods

The following list shows all the methods overridden by the WPFolder class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddFile2Page
- wpAddFile3Page
- wpAddFileMenuPage
- wpRefresh

Class methods

The following list shows all the WPFolder class methods:

- wpclsQueryIconDataN
- wpclsQueryIconN
- wpclsQueryOpenFolders

WPObject class methods

The following list shows all the class methods overridden by the WPFolder class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQuerySettingsPageSize
- wpclsQueryStyle
- wpclsQueryTitle

WPFontPalette

Class definition file: wpfntpal.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPPalette
 WPFontPalette

Description

This is the font palette object class. An instance of this class can be created as a Workplace object. An instance of this class is created initially by the system. It has the title "Font Palette" and resides in the "System Setup" folder.

Instance methods

The following list shows all the WPFontPalette instance methods.

- wpGetFattrsFromPsz

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPFontPalette class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpMenuItemHelpSelected
- wpSetup
- wpSetupOnce

wpSetup override by WPFontPalette

The following table shows the keyname-value pairs supported by the WPFontPalette class.

KEYNAME	VALUE	DESCRIPTION
AUTOSETUP	YES	This specifies that the font palette is to be reinitialized with the default set of fonts.
FONTS	fonttype	These are the initial fonts for each cell in the font palette. The values for each cell are separated by commas. This is equivalent to calling the wpSetupCell method. The fonttype value is presented as the point size followed by a period which is then followed by the face name.
XCELLCOUNT	columns	Number of X cells as decimal digits. If not specified, defaults to 2.
YCELLCOUNT	rows	Number of Y cells as decimal digits. If not specified, defaults to 4.

Example

```
pszSetupString="FONTS=10.Helvetica,"
                     "8.Helvetica,"
                     "10.Courier,"
                     "12.Courier,"
                     "24.Courier,"
                     "10.System";
```

WPPalette instance methods

The following list shows all the methods overridden by the WPFontPalette class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpDragCell
- wpEditCell
- wpPaintCell
- wpQueryPaletteHelp
- wpSelectCell
- wpSetupCell

Class methods

There are currently no class methods defined for the WPFontPalette Workplace object class.

WPObject class methods

The following list shows all the class methods overridden by the WPFontPalette class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryIconData
- wpclsQueryStyle
- wpclsQueryTitle

WPPalette class methods

The following list shows all the class methods overridden by the WPFontPalette class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryEditString

WPHost

Class definition file: wphost.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPFolder
 WPHost

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This is a folder that represents a remote FTP Host. When a FTP Host object is opened, it accesses the specified hostname with the specified username and password.

Instance methods

There are currently no instance methods defined for the WPHost Workplace object class.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to

modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overriden by the WPHost class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectGeneralPage
- wpAddSettingsPages
- wpClose
- wpCreateAnother
- wpCreateFromTemplate
- wpDeleteFromObjUseList
- wpDragOver
- wpDrop
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyPopupMenu
- wpOpen
- wpQueryConfirmations
- wpRestoreState
- wpSaveState
- wpSetup
- wpSetupOnce
- wpUnInitData

wpSetup override by WPHost

The following table shows the keyname-value pairs added by the WPHost class.

KEYNAME	VALUE	DESCRIPTION
HOSTNAME	*hostname*	This sets the hostname to be accessed using an FTP Host object. This value is designated in the "Hostname" field on the Host page. For example: ftp.software.ibm.com
USERNAME	*username*	This sets the username to be used when accessing a hostname using an FTP Host object. This value is designated in the "Username" field on the Host page.
PASSWORD	*password*	This sets the password to be used to access the given host with a given username. This value is designated in the "Password" field on the Host page. This value is not required when the object is created. If one is not specified, the user will be prompted to enter a password when the host is accessed. If specified, passwords are stored in an encrypted form when set.
ACCOUNT	*account*	This sets the account value to be used when accessing a given hostname or username using the FTP Host object. This value is designated in the "Account" field on the Host page. This value is required only when the FTP server being accessed maintains account information for host accesses.
FILETRANSFERTYPE	ASCII	This sets the default file transfer mode for an FTP Host object. This value is designated by selecting the ASCII "Default download type" radio button on Host page 1.

	BINARY	This is the default file transfer mode. This value is designated by selecting the BINARY "Default download type" radio button on Host page 1.
REMOTEDIR	*path*	This specifies which directory will be used as the initial working directory when connecting to a host system using the specified FTP Host object. For example: e:\public\bin, ., / This value is designated in the "Preferred remote directory" field on the Host page 2. The syntax of this path specification must be in a format understood by the remove host's operating system. The given username and account must have permissions set to access this directory on the remote host.
LOCALDIR	*path*	This specifies the directory to be used as the default download directory for GET operations using the FTP Host object if one is not explicitly indicated. This value is designated in the "Preferred local (download) directory" field on Host page 2.
INCLUDE	*pattern*	This is used to filter remote files and directories from the FTP Host object's open views. This value is designated in the "Only display files matching pattern" field on the Include page. The syntax of the pattern must be understood by the remote host's operating system.

WPFileSystem instance methods

The following list shows all the methods overridden by the WPHost class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddFile1Page
- wpAddFile2Page
- wpAddFile3Page

WPFolder instance methods

The following list shows all the methods overridden by the WPHost class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddFirstChild
- wpAddFolderIncludePage
- wpDeleteContents
- wpPopulate

Class methods

There are currently no class methods defined for the WPHost Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPHost class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsCreateDefaultTemplates
- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQueryIconDataN

- wpclsQueryStyle
- wpclsQueryTitle

WPHtml

Class definition file: wphtml.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPDataFile
 WPHtml

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This is the WPHtml object class. An instance of this class can be created as a Workplace object. This class is a kind of WPDataFile that happens to associate itself with a specific file type (HTML). This way, any viewer or editor executable that wants to associate with data files, which have an .HTL or an .HTML file extension or a .TYPE EA of HTML, automatically shows up in a WPHtml object's Open menu. A template of this object is created in the Templates for Internet folder.

Instance methods

There are currently no instance methods defined for the WPHtml Workplace object class.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPHtml class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpCreateAnother
- wpCreateFromTemplate
- wpSetupOnce

Class methods

There are currently no class methods defined for the WPHtml Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPHtml class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsCreateDefaultTemplates
- wpclsInitData
- wpclsQueryDefaultHelp
- wpclsQueryInstanceFilter
- wpclsQueryInstanceType
- wpclsQueryStyle
- wpclsQueryTitle

WPHwManager

Class definition file: wphwmgr.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPFolder
 WPHwManager

Description

This class is specific to Version 4, or higher. of the OS/2 operating system.

This is the Hardware manager folder object class. An instance of this class can be created as a Workplace object. An instance of this class is created initially by the system. It has the title "Hardware Manager" and resides in the "OS/2 System" folder.

Instance methods

The following list shows all the WPHwManager methods:

- wpAddSnoop1Page

WPObject instance methods

The following list shows all the methods overridden by the WPHwManager class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddSettingsPages
- wpCnrInsertObject
- wpInitData
- wpMenuItemSelected
- wpModifyPopupMenu
- wpPopulate
- wpSetupOnce
- wpUnInitData

Class methods

There are currently no class methods defined for the WPHwManager Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPHwManager class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryIconData
- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconDataN
- wpclsQueryStyle
- wpclsQueryTitle
- wpclsUnInitData

WPIcon

Class definition file: wpicon.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPDataFile
 WPIcon

Description

This is the WPIcon object class. All icons are of this class. These icon files are templatable. An instance of this class can be created as a Workplace object. Instances of this class are created initially by the system for each icon in the system.

Instance methods

There are currently no methods defined for the WPIcon Workplace class object.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPIcon class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpSetupOnce
- wpFormatDragItem
- wpQueryIcon

WPDataFile instance methods

The following list shows all the class methods overridden by the WPIcon class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpQueryHandleFromContents
- wpSetContentsFromHandle

Class methods

There are currently no class methods defined for the WPIcon Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPIcon class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryDefaultHelp
- wpclsQueryTitle

WPFileSystem class methods

The following list shows all the class methods overridden by the WPIcon class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryInstanceFilter
- wpclsQueryInstanceType

WPImageFile

Class definition file: wpimage.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPDataFile
 WPImageFile

Description

This is the WPImageFile object class. All data files that contain displayable images are members of this class. This class contains common functions for all data file classes that contain images, such as WPBitmap.

An instance of this class should never be created since this is really the base for other classes.

Instance methods

The following list shows all the WPImageFile methods:

- wpInitData
- wpQueryBitmapData
- wpQueryBitmapHandle
- wpQueryBitmapInfoHeader
- wpReadImageFile
- wpSetBitmapData
- wpWriteImageFile

SOMObject methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPDataFile instance methods

The following list shows all the methods overridden by the WPImageFile class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpQueryHandleFromContents
- wpSetContentsFromHandle

Class methods

There are currently no class methods defined for the WPImageFile Workplace object class.

WPObject class methods

The following list shows all the class methods overridden by the WPImageFile class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryInstanceType
- wpclsQueryStyle
- wpclsQueryTitle

WPJob

Class definition file: wpjob.idl **Class hierarchy**
SOMObject
 WPObject
 WPTransient
 WPJob

Description

This is the job object class. An instance of this class is created by the printer object in its icon or detail view.

Menu ID list

The following list shows the WPPOICM menu IDs for the Job menu for this class:

Menu item text	Menu ID
Menu	WPPOICM_JOB_MENU
Copy	WPPOICM_JOB_COPY
Delete	WPPOICM_JOB_DEL
Status	WPPOICM_JOB_STATUS
Hold	WPPOICM_JOB_HOLD
Release	WPPOICM_JOB_RELEASE
Start Again	WPPOICM_JOB_STARTAGAIN
Print Next	WPPOICM_JOB_PRINTNEXT
Oepn Menu	WPPOICM_JOB_OPENMENU
Content	WPPOICM_JOB_Content

Instance methods

The following list shows all the WPJob methods:

- wpDeleteJob
- wpHoldJob
- wpMoveJobObject
- wpPrintJobNext
- wpQueryJobFile
- wpQueryJobId
- wpQueryJobType
- wpQueryPrintObject
- wpReleaseJob
- wpStartJobAgain

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDumpSelfInt

WPObject instance methods

The following list shows all the methods overridden by the WPJob class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectGeneralPage
- wpAddSettingsPages
- wpAddToObjUseList
- wpConfirmDelete
- wpDeleteFromObjUseList
- wpDragOver
- wpDrop
- wpFilterPopupMenu
- wpFree
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyPopupMenu
- wpOpen
- wpPrintObject
- wpQueryDetailsData
- wpQueryIcon
- wpQueryTitle
- wpSetTitle
- wpUnInitData

Class methods

WPObject class methods

There are currently no class methods defined for the WPJob Workplace class object.

The following list shows all the class methods overridden by the WPJob class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryDetailsInfo
- wpclsQueryIcon
- wpclsQueryStyle
- wpclsQueryTitle

WPKeyboard

Class definition file: wpkeybd.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPKeyboard

Description

This is the keyboard object class. An instance of this class can be created as a Workplace object. An instance of this class is created initially by the system. It has the title "Keyboard" and resides in the "System Setup" folder.

Instance methods

The following list shows all the WPKeyboard methods:

- wpAddKeyboardMappingsPage
- wpAddKeyboardSpecialNeedsPage
- wpAddKeyboardTimingPage

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPKeyboard class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectWindowPage
- wpAddSettingsPages
- wpFilterMenu

Class methods

There are currently no class methods defined for the WPKeyboard Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPKeyboard class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQuerySettingsPageSize
- wpclsQueryStyle
- wpclsQueryTitle

WPAbstract class methods

The following list shows all the class methods overridden by the WPKeyboard class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQuerySetting
- wpclsSetSetting

wpclsSetSettings override by wpKeyboard

The defaults can be specified on a wpclsSetSetting invocation to request that the setting be set to its default value. wpclsQuerySetting, however, never returns an xxxx_DEFAULT value.

Setting	Value	Description
CursorBlinkRate	rate	Sets the rate at which the cursor blinks. This value must be in the range CURSORBLINK_MIN to CURSORBLINK_MAX.
	CURSORBLINK_DEFAULT	Sets the default rate at which the cursor blinks.
EditTitleTextKey	AF_VIRTUALKEY AF_KBDCOMMAND	High word. Sets the key used with another key to edit button text. *Both* of these values are required if the default is not set.

	AF_ALT AF_SHIFT AF_CONTROL	High word. Sets the key used with another key to edit button text. These are optional values.
	VK_F1 through VK_F12 VK_PAGEUP VK_PAGEDOWN VK_HOME VK_END VK_INSERT VK_TAB VK_ENTER	Low word. Sets the key used to edit button text. One of these values is required if the default is not set.
	TEXTEDITKEY_DEFAULT	Sets the default text edit key.
KeyRepeatDelay	rate	Sets the time before the key starts repeating. This value must be in the range REPEATDELAY_MIN to REPEATDELAY_MAX.
	REPEATDELAY_DEFAULT	Sets the default time before the key starts repeating.
KeyRepeatRate	rate	Sets the rate at which a key repeats. This value must be in the range REPEATRATE_MIN to REPEATRATE_MAX.
	REPEATRATE_DEFAULT	Sets the default rate at which a key repeats.
PopupMenuKey	AF_VIRTUALKEY AF_KBDCOMMAND	High word. Sets the key used with another key to display the pop-up menu. *Both* of these values are required if the default is not set.
	AF_ALT AF_SHIFT AF_CONTROL	High word. Sets the key used with another key to display the pop-up menu. These are optional values.
	VK_F1 through VK_F12 VK_PAGEUP VK_PAGEDOWN VK_HOME VK_END VK_INSERT VK_TAB VK_ENTER	Low word. Sets the key used to display the pop-up menu. One of these values is required if the default is not set.
	POPUPKEY_DEFAULT	Sets the default popup menu key.

WPLaunchPad

Class definition file: wplnchpd.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPLaunchPad

Description

This class is specific to version 3, or higher, of the OS/2 operating system. This is the front panel object class.

The Toolbar is a convenience for users that provides fast access to frequently used objects and Desktop actions. Objects can be placed on the Toolbar or in drawers, which are represented by a small button above objects on the Toolbar. There is no limit to the number of Toolbars that can exist in the system. The system Toolbar is defined as the one with an OBJECTID of <WP_LAUNCHPAD>.

An instance of this class can be created as a Workplace object. An instance of this class is created initially by the system. It has the title, "Toolbar" and resides in the "System Startup" folder.

Menu ID list

The following list shows the WPMENUID menu IDs for this class:

Menu item text	Menu ID
Find	WPMENUID_FIND
General help	WPMENUID_EXTENDEDHELP
Palette	WPMENUID_PALETTE

Instance methods

The following list shows all the WPLaunchPad methods.

- wpAddLaunchPadPage1
- wpAddLaunchPadPage2
- wpQueryActionButtons
- wpQueryActionButtonStyle
- wpQueryCloseDrawer
- wpQueryDisplaySmallIcons
- wpQueryDisplayText
- wpQueryDisplayTextInDrawers
- wpQueryDisplayVertical
- wpQueryDrawerHWND
- wpQueryFloatOnTop
- wpQueryHideLaunchPadFrameCtls
- wpQueryObjectList
- wpRefreshDrawer
- wpSetActionButtonStyle
- wpSetCloseDrawer
- wpSetDisplaySmallIcons
- wpSetDisplayText
- wpSetDisplayTextInDrawers
- wpSetDisplayVertical
- wpSetDrawerHWND
- wpSetFloatOnTop
- wpSetHideLaunchPadFrameCtls
- wpSetObjectListFromHObjects
- wpSetObjectListFromObjects
- wpSetObjectListFromStrings

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPLaunchPad class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectWindowPage
- wpAddSettingsPages
- wpDragOver
- wpDrop
- wpFilterMenu
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpOpen
- wpQueryConcurrentView
- wpRestoreState
- wpSaveState
- wpSetup
- wpUnInitData

wpSetup override by WPLaunchPad

Following are keyname-value pairs added by the WPLaunchPad class. The following table contains setup strings that can be used with WinCreateObject and WinSetObjectData to customize Toolbars:

KEYNAME	VALUE	DESCRIPTION
FPOBJECTS	object IDs or path and file names	Adds objects to the end of the Toolbar. If multiple objects exist, the objects are separated by a comma. For example: <WP_OS2WIN>,<WP_OS2EO>
DRAWEROBJECTS	drawer number, object IDs or path and file names	Adds the objects to the end of the numbered Toolbar drawer. The drawer number is followed by a comma-delimited set of object IDs or path and file names. The drawer number and first object must be separated by a comma. Examples of drawer numbers : 0=Toolbar, 1=Left-most drawer, and so forth.
LPCLOSEDRAWER	YES	The Toolbar drawers will close after an object in the drawer is opened.
	NO	The Toolbar drawers will stay open after an object in the drawer is opened.
LPACTIONSTYLE	TEXT	Display the action buttons as text (the default).
	OFF	Turns off the display of action buttons.
	MINI	Displays the action buttons as mini-icons.
	NORMAL	Displays the action buttons as normal (large) icons.
LPVERTICAL	YES	The Toolbar will be displayed vertically.
	NO	The Toolbar will be displayed horizontally (the default).
LPTEXT	YES	The object titles will appear on the Toolbar. This has no effect on the objects in the drawers.
	NO	The object titles will be hidden. This has no effect on the objects in the drawers.
LPDRAWERTEXT	YES	The object titles will appear on the objects in the drawers. This has no effect on the objects on the Toolbar.
	NO	.The object titles will be hidden. This has no effect on the objects on the Toolbar.
LPSMALLICONS	YES	Objects are displayed using small icons.
	NO	Objects are displayed using large (normal) icons.
LPHIDECTLS	YES	The frame controls (title bar and system menu) are hidden (the default).
	NO	The frame controls are displayed.

```
LPFLOAT         YES             The Toolbar will float on top
                                of all other windows.

                NO              The Toolbar will not float on
                                top of all other windows.
```

Class methods

There are currently no class methods defined for the WPLaunchPad Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPLaunchPad class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQuerySettingsPageSize
- wpclsQueryStyle
- wpclsQueryTitle

WPMet

Class definition file: wpmet.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPDataFile
 WPMet

Description

This is the metafile class object. All MET files are of this class. These MET files are templatable. An instance of this class can be created as a Workplace object. Instances of this class are created initially by the system for each MET file in the system.

Instance methods

There are currently no methods defined for the WPMet Workplace class object.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an savestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPMet class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpFormatDragItem
- wpQueryHandleFromContents
- wpSetContentsFromHandle
- wpSetupOnce

Class methods

There are currently no class methods defined for the WPMet Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPMet class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryDefaultHelp
- wpclsQueryTitle

WPFileSystem class methods

The following list shows all the class methods overridden by the WPMet class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryInstanceFilter
- wpclsQueryInstanceType

WPMinWinViewer

Class definition file: wpmwv.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPFolder
 WPMinWinViewer

Description

This class is specific to version 3, or higher, of the OS/2 operating system.

This is the minimized window viewer object class. This is a special folder that contains only minimized window objects. An instance of this class can be created as a Workplace object. An instance of this class is created by the system during initialization.

Menu ID list

The following list shows the WPMENUID menu IDs for this class:

Menu item text	Menu ID
Close	WPMENUID_CLOSE

Instance methods

The following list shows all the WPMinWinViewer methods.

- wpFindMinWindow

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPMinWinViewer class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectWindowPage
- wpDragOver
- wpFilterMenu
- wpMenuItemSelected
- wpModifyMenu
- wpOpen

WPFileSystem instance methods

The following list shows all the methods overridden by the WPMinWinViewer class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddFileMenuPage

WPFolder instance methods

The following list shows all the methods overridden by the WPMinWinViewer class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddFolderIncludePage
- wpAddFolderSelfClosePage
- wpAddFolderView2Page
- wpAddFolderView3Page
- wpQueryFldrSortClass

Class methods

There are currently no class methods defined for the WPMinWinViewer Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPMinWinViewer class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQueryIconDataN
- wpclsQuerySettingsPageSize
- wpclsQueryStyle
- wpclsQueryTitle

WPMouse

Class definition file: wpmouse.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPMouse

Description

This is the mouse object class. An instance of this class can be created as a Workplace object. An instance of this class is created initially by the system. It has the title "Mouse" and resides in the "System Setup" folder.

Instance methods

The following list shows all the WPMouse methods:

- wpAddMouseCometPage
- wpAddMouseMappingsPage
- wpAddMousePtrPage
- wpAddMouseTimingPage
- wpAddMouseTypePage

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPMouse class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectWindowPage
- wpAddSettingsPages
- wpFilterMenu

Class methods

There are currently no class methods defined for the WPMouse Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPMouse class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQueryStyle
- wpclsQueryTitle

WPAbstract class methods

The following list shows all the class methods overridden by the WPMouse class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQuerySetting
- wpclsSetSetting

wpclsSetSettings override by wpMouse

The defaults can be specified on a wpclsSetSetting invocation to request that the setting be set to its default value. wpclsQuerySetting, however, never returns an xxxx_DEFAULT value.

Setting	Value	Description
ButtonSetup	BUTTONS_LEFTHANDED	Sets the mouse buttons for a left-handed person.
	BUTTONS_RIGHTHANDED	Sets the mouse buttons for a right-handed person.
	BUTTONS_DEFAULT	Sets the default setup for the mouse buttons.
DisplayWindowListButton	INP_NONE	High word. This value is required if the default is not set.
	WM_BUTTON2CLICK WM_CHORD	Low word. Sets the button which displays the window list. One of these values is required if the default is not set.
	WINDOWLISTBUTTON_DEFAULT	Sets the default button which displays the window list.
DoubleClickTime	time	Sets the time between mouse clicks. Higher values allow more time between mouse clicks, where as lower values allow less time between mouse clicks. This value must be in the range 170 to 1 060.

	DOUBLECLICK_DEFAULT	Sets the default time between mouse clicks.
DragObjectButton	INP_NONE	High word. This value is required if the default is not set.
	WM_BUTTON1MOTIONSTART WM_BUTTON2MOTIONSTART	Low word. Sets the button which drags objects. One of these values is required if the default is not set.
	DRAGBUTTON_DEFAULT	Sets the button which drags objects to the default.
EditTitleTextButton	INP_ALT INP_CTRL INP_NONE INP_SHIFT	High word. Sets the key used with a mouse button to edit button text. One of these values is required if the default is not set.
	WM_BUTTON1CLICK WM_BUTTON2CLICK WM_BUTTON1DBLCLK WM_BUTTON2DBLCKL	Low word. Sets the mouse button used to edit button text. One of these values is required if the default is not set.
	TEXTEDITBUTTON_DEFAULT	Sets the text edit button to the default.
PopupMenuButton	INP_ALT INP_CTRL INP_NONE INP_SHIFT	High word. Sets the key used with a mouse button to display pop-up menus. One of these values is required if the default is not set.
	WM_BUTTON1CLICK WM_BUTTON2CLICK WM_BUTTON1DBLCLK WM_BUTTON2DBLCKL WM_CHORD	Low word. Sets the mouse button used to display pop-up menus. One of these values is required if the default is not set.
	POPUPBUTTON_DEFAULT	Sets the default key and mouse button used to display pop-up menus.
TrackingSpeed	speed	Sets the mouse tracking speed. This value must be in the range 1 to 7, where 1

```
                                    is the fastest
                                    value.

TRACKING_DEFAULT                    Sets the default
                                    mouse tracking
                                    speed.
```

WPNetgrp

Class definition file: wpnetgrp.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPFolder
 WPNetgrp

Description

This class is specific to Version 3, or higher, of the OS/2 operating system.

This is the network group object class. An instances of this class can be created as a Workplace object. Instances of this class are created for every LAN system in a WPNetwork, for example LAN server and Novell**).

Menu ID list

The following list shows the WPMENU menu IDs for this class:

Menu item text	Menu ID
Login...	WPMENUID_LOGIN
Logout...	WPMENUID_LOGOUT
Refresh now	WPMENUID_REFRESH

Instance methods

The following list shows all the WPNetgrp methods.

- wpAddNetworkPage
- wpQueryNetIdentity
- wpSetNetIdentity

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPNetgrp class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectGeneralPage
- wpAddObjectGeneralPage2
- wpAddSettingsPages
- wpFilterMenu
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyMenu
- wpOpen
- wpRestoreState
- wpSaveState
- wpSetup
- wpUnInitData

wpSetup override by WPNetGrp

The following table shows the keyname-value pairs added by the WPNetGrp class.

KEYNAME	VALUE	DESCRIPTION
NETGCOMMENT	title	This sets the title for a network group folder.

WPFileSystem instance methods

The following list shows all the methods overridden by the WPNetgrp class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddFile1Page
- wpAddFile2Page
- wpAddFile3Page

WPFolder instance methods

The following list shows all the methods overridden by the WPNetgrp class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddFolderIncludePage
- wpDeleteContents
- wpPopulate

Class methods

There are currently no class methods defined for the WPNetgrp Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPNetgrp class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQueryIconDataN
- wpclsQuerySettingsPageSize
- wpclsQueryStyle
- wpclsQueryTitle

WPNetLink

Class definition file: wpnetlnk.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPShadow
 WPNetLink

Description

This class is specific to Version 3, or higher, of the OS/2 operating system.

This is the network link object class. This object handles all saving and restoring of data for the shared directory object by providing a persistent link or reference to that shared directory. The purpose of this link is to locally store all the instance data of the shared directory that this object is linked to and then reroute all requests for help, context menus, open views, and so forth, to this object. An instance of this class can be created as a Workplace object.

Instance methods

The following list shows all the WPNetLink methods.

- wpQueryObjectNetId
- wpSetObjectNetId

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPNetLink class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpCreateShadowObject
- wpFilterMenu
- wpFilterPopupMenu
- wpInitData
- wpMenuItemHelpSelected
- wpModifyMenu
- wpModifyPopupMenu
- wpQueryStyle
- wpRestoreState
- wpSaveState
- wpUnInitData

WPShadow instance methods

The following list shows all the methods overridden by the WPNetLink class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpSetLinkToObject

Class methods

There are currently no class methods defined for the WPNetLink Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPNetLink class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryStyle
- wpclsQueryTitle

WPNetwork

Class definition file: wpnetwrk.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPFolder
 WPNetwork

Description

This class is specific to Version 3, or higher, of the OS/2 operating system.

This is the network object class. An instances of this class can be created as a Workplace object. This object is the "network" folder that is created by the system on the Desktop. It is visible only if the LAN is attached to the machine being used.

Menu ID list

The following list shows the WPMENUID menu IDs for this class:

Menu item text	Menu ID
Refresh now	WPMENUID_REFRESH

Instance methods

There are currently no methods defined for the WPNetwork Workplace object class.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPNetwork class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpFilterMenu
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyMenu
- wpOpen
- wpRestoreState
- wpSetup

WPFileSystem instance methods

The following list shows all the methods overridden by the WPNetwork class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddFile1Page
- wpAddFile2Page
- wpAddFile3Page

WPFolder instance methods

The following list shows all the methods overridden by the WPNetwork class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddFolderIncludePage
- wpPopulate

Class methods

There are currently no class methods defined for the WPNetwork Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPNetwork class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQueryIconDataN
- wpclsQueryStyle
- wpclsQueryTitle

WPObject

Class definition file: wpobject.idl

Class hierarchy
SOMObject
 WPObject

Description

This is the root Workplace object class. This is the fundamental class from which all Workplace objects are derived, regardless of where they are actually stored. Immediate descendant classes of WPObject are called *storage classes* , because they take responsibility for storing the object information, typically in a persistent form. Predefined Workplace object storage classes are the following:

- WPAbstract
- WPFileSystem
- WPTransient

A Workplace object of this class cannot be created.

Menu ID list

The following list shows the WPMENUID menu IDs for this class:

Menu item text	Menu ID
Close	WPMENUID_CLOSE
Copy	WPMENUID_COPY
Create another	WPMENUID_CREATEANOTHER
Create shadow	WPMENUID_CREATESHADOW
Delete	WPMENUID_DELETE
Move	WPMENUID_MOVE
Open	WPMENUID_OPEN
Print	WPMENUID_PRINT
Settings	WPMENUID_PROPERTIES
General help	WPMENUID_EXTENDEDHELP
Help	WPMENUID_HELP
Help index	WPMENUID_HELPINDEX
Keys help	WPMENUID_HELPKEYS
Using help	WPMENUID_HELP_FOR_HELP
Cancel drag	WPMENUID_PUTDOWN_CANCEL
Copy	WPMENUID_PUTDOWN_COPY
Create from template	WPMENUID_PUTDOWN_CREATE
Create shadow	WPMENUID_PUTDOWN_LINK
Drop	WPMENUID_PUTDOWN
Move	WPMENUID_PUTDOWN_MOVE
Pickup	WPMENUID_PICKUP

Instance methods

The following lists show how all the WPObject instance methods are related within functional areas:

Settings Notebook methods

- wpAddObjectGeneralPage
- wpAddObjectGeneralPage2
- wpAddObjectWindowPage
- wpAddSettingsPages
- wpInsertSettingsPage

Save/Restore State methods

- wpRestoreData
- wpRestoreLong
- wpRestoreState
- wpRestoreString
- wpSaveData
- wpSaveDeferred
- wpSaveImmediate
- wpSaveLong
- wpSaveState
- wpSaveString

Object Usage methods

- wpAddToObjUseList
- wpAssertObjectMutexSem
- wpCnrDeleteUseItem
- wpDeleteFromObjUseList
- wpFindUseItem
- wpIsLocked
- wpLockObject

- wpReleaseObjectMutexSem
- wpRequestObjectMutexSem
- wpUnlockObject

Pop-Up Menu methods

- wpAppendObject
- wpClose
- wpCopyObject
- wpCreateAnother
- wpCreateFromTemplate
- wpCreateShadowObject
- wpCreateShadowObjectExt
- wpDelete
- wpDisplayHelp
- wpDisplayMenu
- wpFilterMenu
- wpFilterPopupMenu
- wpHide
- wpInsertMenuItems
- wpInsertPopupMenuItems
- wpIsObjectInitialized
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyMenu
- wpModifyPopupMenu
- wpMoveObject
- wpObjectReady
- wpOpen
- wpPrintObject
- wpQueryMenuStyle
- wpReplaceObject
- wpRestore
- wpSetMenuStyle

Query and Set Object Information methods

- wpFindTaskRec
- wpIdentify
- wpModifyStyle
- wpQueryButtonAppearance
- wpQueryConcurrentView
- wpQueryConfirmations
- wpQueryContainerFlagPtr
- wpQueryCoreRecord
- wpQueryDefaultHelp
- wpQueryDefaultIconPos
- wpQueryDefaultView
- wpQueryDetailsData
- wpQueryFolder
- wpQueryHandle
- wpQueryIcon
- wpQueryIconData
- wpQueryMinWindow
- wpQueryNameClashOptions
- wpQueryObjectID
- wpQueryScreenGroupID (Intel)
- wpQueryStyle
- wpQueryTrueStyle
- wpQueryTitle
- wpSetButtonAppearance
- wpSetConcurrentView
- wpSetDefaultHelp
- wpSetDefaultIconPos
- wpSetDefaultView
- wpSetFolder
- wpSetIcon
- wpSetIconData
- wpSetMinWindow
- wpSetObjectID
- wpSetStyle
- wpSetTaskRec
- wpSetTitle
- wpViewObject

Error Handling methods

- wpQueryError
- wpSetError

Memory Management methods

- wpAllocMem
- wpFreeMem

Direct Manipulation methods

- wpDraggedOverObject
- wpDragOver
- wpDrop
- wpDroppedOnObject
- wpEndConversation
- wpFormatDragItem
- wpRender
- wpRenderComplete

Miscellaneous methods

- wpCnrInsertObject
- wpCnrRefreshDetails
- wpCnrRemoveObject
- wpCnrSetEmphasis
- wpConfirmDelete
- wpConfirmObjectTitle
- wpCopiedFromTemplate
- wpDoesObjectMatch
- wpFindViewItem
- wpIsDeleteable
- wpRegisterView
- wpSwitchTo
- wpWaitForClose

Setup/Cleanup methods

- wpFree
- wpInitData
- wpScanSetupString
- wpSetup
- wpSetupOnce
- wpUnInitData

wpSetup override by WPObject

The following table shows the keyname-value pairs added by the WPObject class.

KEYNAME	VALUE	DESCRIPTION
CCVIEW	DEFAULT	The default value of the concurrent view setting of the system is used when the user selects open.
	YES	This sets the object's multiple concurrent view behavior, so that new views of this object are created every time the user selects open.
	NO	Opened view of this object resurfaces when the user selects open.
DEFAULTVIEW	SETTINGS	This sets the default open view to Settings view.
	id	This sets the default open view to the id of a user-added view (0-9).
	DEFAULT	This sets the default open view to the object's class default view, as returned by wpclsQueryDefaultView.

		This is the default value.
HELPLIBRARY	filename	This sets the help library.
HELPPANEL	id	This sets the object's default help panel for object-specific help. This is equivalent to calling wpSetDefaultHelp.
HIDEBUTTON	YES	Views of this object have a hide button as opposed to a minimize button.
	NO	Views of this object have a minimize button as opposed to a hide button. The default is the current system Button appearance for windows setting. This cannot be specified here.
ICONFILE	filename	This sets the object's icon. This is equivalent to calling wpSetIconData.
ICONPOS	x,y	This sets the object's initial icon position in a folder. The x and y values represent the position in the object's folder in percentage coordinates.
ICONRESOURCE	id,module	This sets the object's icon. This is equivalent to calling wpSetIconData. The "id" is the icon resource ID in the dynamic link library (DLL) "module".
LOCKEDINPLACE	YES	This sets the object's icon locked in place property, so that the object's icon is fixed in position in an open icon view of the folder containing the object.
	NO	This resets the object's icon locked in place property, so that the object's icon position can be moved.
MENUITEMSELECTED	menu-item-ID	This simulates selecting the specified menu item from the object's pop-up menu.
MENUS	This sets the object's pop-up menu property to short menus.	
	SHORT	This sets the object's pop-up menu property to long menus. This is the default value.
MINWIN	HIDE	Views of this object hide when their minimize button is selected.
	VIEWER	Views of this object minimize to the minimized window viewer when their minimize button is selected.
	DESKTOP	Views of this object minimize to the desktop when their minimize button

		is selected. The default is the current Minimize button setting. This cannot be specified here.
NOCOPY	YES	This sets the object's no copy property, so that the object cannot be copied. This is equivalent to calling wpSetStyle with a style of OBJSTYLE_NOCOPY.
	NO	This resets the object's no copy property, so that the object can be copied. This is the default value.
NODELETE	YES	This sets the object's no delete property, so that the object cannot be deleted. This is equivalent to calling wpSetStyle with a style of OBJSTYLE_NODELETE.
	NO	This resets the object's no delete property, so that the object can be deleted. This is the default value.
NODRAG	YES	This sets the object's no drag property, so that the object cannot be dragged. This is equivalent to calling wpSetStyle with a style of OBJSTYLE_NODRAG.
	NO	This resets the object's no drag property, so that the object can be dragged. This is the default value.
NODROP	YES	This sets the object's no drop property, so that no other object can be dropped on it. This is equivalent to calling wpSetStyle with a style of OBJSTYLE_NODROPON.
	NO	This resets the object's no drop property, so that another object can be dropped on it. This is the default value.
NOLINK	YES	This sets the object's no link property, so that the object cannot be linked. This is equivalent to calling wpSetStyle with a style of OBJSTYLE_NOLINK.
	NO	This resets the object's no link property, so that the object can be linked. This is the default value. This is the same as the NOSHADOW keyword.
NOMOVE	YES	This sets the object's no move property, so that the object cannot be moved. This is equivalent to calling wpSetStyle with a style of OBJSTYLE_NOMOVE.
	NO	This resets the object's no move property, so that the object can be moved.

		This is the default value.
NOPRINT	YES	This sets the object's no print property, so that the object cannot be printed. This is equivalent to calling wpSetStyle with a style of OBJSTYLE_NOPRINT.
	NO	This resets the object's no print property, so that the object can be printed. This is the default value.
NORENAME	YES	This sets the object's no rename property, so that the object cannot be renamed. This is equivalent to calling wpSetStyle with a style of OBJSTYLE_NORENAME.
	NO	This resets the object's no rename property, so that the object can be renamed. This is the default value.
NOSETTINGS	YES	This sets the object's no settings property, so that the object' settings cannot be opened. This is equivalent to calling wpSetStyle with a style of OBJSTYLE_NOSETTINGS.
	NO	This resets the object's no settings property, so that the object's settings can be opened. This is the default value.
NOSHADOW	YES	This sets the object's no link property, so that the object cannot be linked. This is equivalent to calling wpSetStyle with a style of OBJSTYLE_NOLINK.
	NO	This resets the object's no link property, so that the object can be linked. This is the default value. This is the same as the NOLINK keyword.
NOTVISIBLE	YES	This sets the object's not visible property, so that the object cannot be made visible. This is equivalent to calling wpSetStyle with a style of OBJSTYLE_NOTVISIBLE.
	NO	This resets the object's not visible property, so that the object can be made visible. This is the default value.
OBJECTID	<name>	This sets a persistent ID for the object. The OBJECTID stays with the object even if it is moved or renamed. The object pointer or handle can be retrieved via wpclsQueryObject or WinQueryObject , respectively. An OBJECTID is any unique string preceded with a "<" and terminated with a ">".

		The default is to not have an ID.
OPEN	SETTINGS	This opens the Settings view when an object is created or when WinSetObjectData is called.
	DEFAULT	This opens the default view when an object is created or when WinSetObjectData is called. The default is to not open a view.
TEMPLATE	YES	This sets the object's template property. This is equivalent to calling wpSetStyle with a style of OBJSTYLE_TEMPLATE.
	NO	This resets the object's template property. This is the default value.
TITLE	Title	This sets the object's title. This is equivalent to calling wpSetTitle.

Example

```
pszSetupString="TITLE=MYObject;"
               "ICONFILE=myobj.ico;"
               "HELPPANEL=132;"
               "TEMPLATE=YES;"
               "NODELETE=NO";
```

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct
- somInit
- somUninit

Class Methods

The following list shows all the WPObject class methods:

- wpclsCreateDefaultTemplates
- wpclsDecUsage
- wpclsFindObjectEnd
- wpclsFindObjectFirst
- wpclsFindObjectNext
- wpclsFindOneObject
- wpclsIncUsage
- wpclsInitData
- wpclsInsertMultipleObjects
- wpclsMakeAwake
- wpclsNew
- wpclsObjectFromHandle
- wpclsQueryButtonAppearance
- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryDetails
- wpclsQueryDetailsInfo
- wpclsQueryError
- wpclsQueryExtendedCriteria
- wpclsQueryFolder
- wpclsQueryIcon
- wpclsQueryIconData
- wpclsQueryObject

- wpclsQuerySearchInfo
- wpclsQuerySettingsPageSize
- wpclsQueryStyle
- wpclsQueryTitle
- wpclsRemoveObjects
- wpclsSetError
- wpclsSetIconData
- wpclsSetIcon
- wpclsSetSettingsPageSize
- wpclsUnInitData

SOMClass methods

The following list shows all the class methods overridden by the the WPObject class. These methods are overridden to modify the behavior defined by an ancestor class.

- somClassReady
- somNew
- somRenew

WPPalette

Class definition file: wppalet.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPPalette

Description

This is the palette object class. There are no instances of this class initially created by the system.

Instance methods

The following list shows all the WPPalette instance methods:

- wpDragCell
- wpEditCell
- wpFreeCellData
- wpInitCellStructs
- wpPaintCell
- wpPaintPalette
- wpQueryPaletteHelp
- wpQueryPaletteInfo
- wpRedrawCell
- wpRestoreCellData
- wpSaveCellData
- wpSelectCell
- wpSetPaletteInfo
- wpSetupCell

SOMObject instance methods

The following list shows all the instance methods overridden by the WPPalette class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPPalette class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpFilterMenu
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected

- wpModifyMenu
- wpOpen
- wpRestoreState
- wpSaveState
- wpSetup
- wpUnInitData

wpSetup override by WPPalette

The following table shows the keyname-value pairs added by the WPPalette class.

KEYNAME	VALUE	DESCRIPTION
XCELLCOUNT	columns	Number of columns of cells. If not specified, defaults to 13.
YCELLCOUNT	rows	Number of rows of cells. If not specified, defaults to 9.
XCELLWIDTH	width	Width in dialog units of each cell.
YCELLHEIGHT	height	Height in dialog units of each cell.
XCELLGAP	gap	X separation in dialog units between each cell.
YCELLGAP	gap	Y separation in dialog units between each cell.

Example

```
pszSetupString="XCELLCOUNT=3,"
               "YCELLCOUNT=4";
```

Class methods

The following list shows all the WPPalette class methods:

- wpclsQueryEditString

WPObject class methods

The following list shows all the class methods overridden by the WPPalette class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryStyle
- wpclsQueryTitle

WPPdr

Class definition file: wppdr.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient
 WPPdr

Description

This is the printer driver object class. An instance of this class is created by the print object in its Settings view.

Instance methods

There are currently no methods defined for the WPPdr Workplace class object.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPPdr class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDumpSelfInt

WPObject instance methods

The following list shows all the methods overridden by the WPPdr class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddSettingsPages
- wpConfirmDelete
- wpDragOver
- wpDrop
- wpFilterPopupMenu
- wpFree
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyPopupMenu
- wpOpen
- wpQueryIcon
- wpQueryTitle
- wpSetup
- wpUnInitData

wpSetup override by WPPdr

The following table shows the keyname-value pairs added by the WPPdr class.

KEYNAME	VALUE	DESCRIPTION
INSTPATH	pathname	This sets the name of the install directory. When the printer driver object is created, and the driver is not already installed, the required driver will be installed from this path.
PORTNAME	portname	This sets the port name required by the Windows printer driver. Portname may be one of the following values: LPT1 COM1 COM4 LPT2 COM2 FILE LPT3 COM3 The WINOS2 keyname should also be specified when using this keyname.
PRINTDRIVER	driver name	This sets the full driver name for the printer driver object. A full driver name is in the driver.device format. For example: IBM42XXX.IBM 4201 Proprinter III
PROMPT	YES	This specifies that the user should be prompted for installation diskettes. When creating a printer driver object, and the required driver is not already installed, the user will be prompted for printer driver installation diskettes.
	NO	This specifies that the user should not be prompted

```
                                    for installation diskettes.

WINOS2                  YES         This specifies that the
                                    WINOS2 (or Windows) driver
                                    should also be installed
                                    when creating the OS/2
                                    printer driver object. The
                                    PORTNAME keyname should
                                    also be specified when
                                    using this keyname.

                        NO          This specifies that the
                                    WINOS2 driver should not be
                                    installed.
```

Class methods

There are currently no class methods defined for the WPPdr Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPPdr class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIcon
- wpclsQueryStyle
- wpclsQueryTitle

WPPif

Class definition file: wppif.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPDataFile
 WPPif

Description

This is the Picture Interchange Format (PIF) class object. All PIF files are of this class. These PIF files are templatable. An instance of this class can be created as a Workplace object. Instances of this class are created initially by the system for each PIF file in the system.

Instance methods

There are currently no methods defined for the WPPif Workplace class object.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPPif class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpSetupOnce

Class methods

There are currently no class methods defined for the WPPif Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPPif class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryDefaultHelp
- wpclsQueryTitle

WPFileSystem class methods

The following list shows all the class methods overridden by the WPPif class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryInstanceFilter
- wpclsQueryInstanceType

WPPointer

Class definition file: wpptr.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPDataFile
 WPPointer

Description

This is the WPPointer class. All pointers are of this class. These pointer files are templatable. An instance of this class can be created as a Workplace object. Instances of this class are created initially by the system for each pointer in the system.

Instance methods

There are currently no methods defined for the WPPointer Workplace class object.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPPointer class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpFormatDragItem
- wpQueryPointerFromContents
- wpQueryIcon
- wpSetContentsFromHandle
- wpSetupOnce

Class methods

There are currently no class methods defined for the WPPointer Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPPointer class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryDefaultHelp
- wpclsQueryTitle

WPFileSystem class methods

The following list shows all the class methods overridden by the WPPointer class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryInstanceFilter
- wpclsQueryInstanceType

WPPort

Class definition file: wpport.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient
 WPPort

Description

This is the port object class. An instance of this class is created by the print object in its Settings view.

Instance methods

There are currently no methods defined for the WPPort Workplace class object.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDumpSelfInt

WPObject instance methods

The following list shows all the methods overridden by the WPPort class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpCnrSetEmphasis
- wpConfirmDelete
- wpDragOver
- wpDrop
- wpFilterPopupMenu
- wpFree
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyPopupMenu
- wpOpen
- wpQueryDetailsData
- wpQueryIcon
- wpSetup
- wpQueryTitle
- wpUnInitData

Class methods

There are currently no class methods defined for the WPPort Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPPort class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryDetailsInfo
- wpclsQueryIcon
- wpclsQueryIconData
- wpclsQueryStyle
- wpclsQueryTitle

WPPower

Class definition file: wppower.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPPower

Description

This class is specific to version 2.1, or higher, of the OS/2 operating system.

This is the power object class that represents the power management support. An instance of this class can be created as a Workplace object. An instance of this class is created initially by systems that support power management. It cannot be deleted. It has the title "Power" and resides in the "System Setup" folder.

Instance methods

The following list shows all the WPPower methods:

- wpAddPowerPage
- wpAddPowerViewPage
- wpChangePowerState
- wpQueryAutoRefresh
- wpQueryDefStatusView
- wpQueryPowerConfirmation
- wpQueryPowerManagement
- wpQueryRefreshRate
- wpSetAutoRefresh
- wpSetDefStatusView
- wpSetPowerConfirmation
- wpSetPowerManagement
- wpSetRefreshRate

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPPower class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectWindowPage
- wpAddSettingsPages
- wpFilterMenu
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyMenu
- wpOpen
- wpRestoreState
- wpSaveState

Class methods

There are currently no class methods defined for the WPPower Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPPower class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView

- wpclsQueryIconData
- wpclsQueryStyle
- wpclsQueryTitle

WPPrinter

Class definition file: wpprint.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPPrinter

Description

This is the printer object class. An instance of this class can be created as a Workplace object. An instance of this class is created initially by the system in its template form. It has the title "Printer" and resides in the "Templates" folder. Instances of this class are also created initially by the system for each print configured. Each instance has a title corresponding to the description of the configured queue and printer and resides on the desktop.

Instance methods

The following list shows all the WPPrinter methods:

- wpDeleteAllJobs
- wpHoldPrinter
- wpJobAdded
- wpJobChanged
- wpJobDeleted
- wpQueryComputerName
- wpQueryPrinterName
- wpQueryQueueOptions
- wpQueryRemoteOptions
- wpReleasePrinter
- wpSetComputerName
- wpSetJobProperties
- wpSetDefaultPrinter
- wpSetPrinterName
- wpSetQueueOptions
- wpSetRemoteOptions

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDumpSelfInt

WPObject instance methods

The following list shows all the methods overridden by the WPPrinter class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddSettingsPages
- wpConfirmDelete
- wpCopiedFromTemplate
- wpCopyObject
- wpCreateAnother
- wpCreateFromTemplate
- wpCreateShadowObject
- wpDragOver
- wpDrop
- wpFilterPopupMenu
- wpFree
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyPopupMenu
- wpMoveObject

- wpOpen
- wpQueryDefaultHelp
- wpQueryDefaultView
- wpQueryDetailsData
- wpQueryIcon
- wpQueryIconData
- wpQueryStyle
- wpQueryTitle
- wpRestoreState
- wpSaveState
- wpSetDefaultView
- wpSetIcon
- wpSetIconData
- wpSetTitle
- wpSetup
- wpUnInitData

wpSetup override by WPPrint

The following table shows the keyname-value pairs added by the WPPrint class.

KEYNAME	VALUE	DESCRIPTION
APPDEFAULT	YES	This printer object is to become the application's default printer object for printing.
	NO	This printer object is not to become the application's default printer object for printing.
DEFAULTVIEW	DETAILS	Default open view for this printer object is in details view.
	ICON	Default open view for this printer object is in icon view.
JOBDIALOGBEFOREPRINT	YES	The job properties dialog is displayed before printing.
	NO	The job properties dialog is not displayed before printing.
JOBPROPERTIES	filename	The complete path to a binary file containing the default job properties for this printer object. This file can be created by saving the PRQINFO3->pDriverData data to a file; this data can be obtained by using the SplQueryQueue API of the spooler. For more information about spooler functions, see the *Presentation Manager Programming Reference*.
OUTPUTTOFILE	YES	The output of this printer object goes to a file. The user will be prompted for a file name each time a print job is submitted to this printer object.
	NO	The output of this printer object does not go to a file.
PORTNAME	portname	The names of already installed ports to which this printer object is to be attached. In the case

		of more than one port, specify a comma-separated list.
PRINTDRIVER	driver.device	The complete name of the print-driver that this printer object is to use. For example: 'IBM42XX.IBM 420 Proprinter III' In the case of more than one print driver, specify a comma-separated list. These printer drivers must already be installed.
PRINTERSPECIFICFORMAT	YES	The printer object spools print jobs in PM_Q_RAW format.
	NO	The printer object spools print jobs in PM_Q_STANDARD format.
PRINTWHILESPOOLING	YES	The printing is enabled while the job is spooling.
	NO	The printing is disabled while the job is spooling.
PRIORITY	priority number (0-9)	This sets the priority of the print queue associated with the printer object. All print jobs that are submitted to the printer object will inherit a job priority derived from the print queue's priority. Zero is the highest priority and 9 is the lowest priority.
QSTARTTIME	time	The time when the printer object starts printing. The time format is HH:MM, and the base is a 24-hour clock.
QSTOPTIME	time	The time when the printer object is to stop printing. The time format is HH:MM, and the base is a 24-hour clock.
QUEUENAME	name	The local queue name for the printer object. If a queue name is not specified, one is created by the printer object. The QUEUENAME key will be ignored if this object has already been assigned a queue.
QUEUEPARMS	qparm	This sets the queue parameters, and can be any text string. Queue parameters are specific instructions that are read and used by OS/2 print drivers, OS/2 queue drivers, and the OS/2 spooler.
QUEUEDRIVER	qdrvname	The queue driver name. The queue driver must already be installed.
SEPARATORFILE	filename	A separator file that prints before each print job.
SYNCJOBPROP	YES	This specifies that the default job properties of

		the printer object are to be synchronized with the network's printer job properties.
	NO	This specifies that the default job properties will not be synchronized.
SYNCPRINTERPROP	YES	This specifies that the printer properties of the printer object are to be synchronized with the network's printer job properties.
	NO	This specifies that the printer properties will not be synchronized.
TAKEDEFAULTS	YES	This specifies that the printer object takes the default values when it is created.
	NO	This specifies that the printer object does not take default values when it is created. This will cause the "Create a printer" dialog to be displayed.

Class methods

There are currently no class methods defined for the WPPrinter Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPPrinter class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryDetailsInfo
- wpclsQueryIcon
- wpclsQueryIconData
- wpclsQuerySettingsPageSize
- wpclsQueryStyle
- wpclsQueryTitle

WPProgram

Class definition file: wppgm.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPProgram

Description

This is the program object class. This class provides an object that points to executable programs and allows the user to run that program by simply double-clicking on the program object. The program can also contain a variety of useful additional parameters, such as the environment for the program and the parameters that are passed to it. An instance of this class can be created as a Workplace object and is created initially by the system in its template form. It has the title "Program" and resides in the "Templates" folder.

Other instances of this class initially created by the system include:

- "DOS Full Screen" in the "Command Prompts" folder
- "DOS Window" in the "Command Prompts" folder
- "OS/2 Full Screen" in the "Command Prompts" folder
- "OS/2 Window" in the "Command Prompts" folder
- Every object in the "Games" folder
- Some objects in the "Information" folder
- Every object in the "Productivity" folder

Menu ID list

The following list shows the WPMENUID menu IDs for this class:

Menu item text	Menu ID
Program	WPMENUID_PROGRAM

Instance methods

The following list shows all the WPProgram methods:

- wpAddProgramAssociationPage
- wpAddProgramEnvironmentPage
- wpAddProgramPage
- wpAddProgramSessionPage
- wpQueryAssociationFilter
- wpQueryAssociationType
- wpQueryProgDetails
- wpQueryProgramAssociations
- wpQueryWarnBeforeStart (WPProgram)
- wpSetAssociationFilter
- wpSetAssociationType
- wpSetProgDetails
- wpSetProgramAssociations
- wpSetProgIcon
- wpSetWarnBeforeStart (WPProgram)

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPProgram class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddSettingsPages
- wpCopyObject
- wpCreateAnother
- wpCreateFromTemplate
- wpDragOver
- wpDrop
- wpFilterMenu
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyMenu
- wpMoveObject
- wpOpen
- wpQueryDefaultHelp
- wpQueryIconData
- wpRestoreState
- wpSaveState
- wpSetIconData
- wpSetTitle
- wpSetup
- wpUnInitData

wpSetup override by WPProgram

The following table shows the keyname-value pairs added by the WPProgram class.

KEYNAME	VALUE	DESCRIPTION

ASSOCFILTER	filters	Sets the filename filter for files associated with this program. Multiple filters are separated by commas.
ASSOCTYPE	type	Sets the type of files associated with this program. Multiple types are separated by commas.
EXENAME	filename	Sets the name of the program.
MAXIMIZED	YES	The program window is maximized upon startup.
	NO	The program window is restored to normal upon startup. This is the default setting.
MINIMIZED	YES	The program window is minimized upon startup.
	NO	The program window is restored to normal upon startup. This is the default setting.
PROGTYPE	DOSMODE	Sets the session type to stand-alone DOS mode.
	FULLSCREEN	Sets the session type to OS/2 full-screen mode.
	PM	Sets the session type to PM.
	PROG_30_STD	Sets the session type to standard compatibility full screen mode.
	PROG_31_ENH	Sets the session type to enhanced compatibility full screen mode.
	PROG_31_ENHSEAMLESSCOMMON	Sets the session type to WIN-OS/2* window in the WIN-OS/2 enhanced compatibility common session.
	PROG_31_ENHSEAMLESSVDM	Sets the session type to WIN-OS/2 window in a separate session enhanced compatibility mode.
	PROG_31_STD	Sets the session type to standard compatibility full screen mode.
	PROG_31_STDSEAMLESSCOMMON	Sets the session type to WIN-OS/2 window in the WIN-OS/2 standard compatibility common session.
	PROG_31_STDSEAMLESSVDM	Sets the session type to WIN-OS/2 window in a separate session standard compatibility mode.
	SEPARATEWIN	Sets the session type to WIN-OS/2 window running in a separate VDM.
	VDM	Sets the session type to DOS full-screen.
	WIN	Sets the session type to WIN-OS/2 full screen.
	WINDOWABLEVIO	Sets the session type to OS/2 windowed.
	WINDOWEDVDM	Sets the session type to DOS windowed.

	WINDOWEDWIN	Sets the session type to WIN-OS/2 window.
NOAUTOCLOSE	YES	Leaves the window open when the program terminates.
	NO	Closes the window when the program terminates. This is the default setting.
PARAMETERS	params	Sets the parameters list, which may include substitution characters.
SET	XXX=VVV	XXX is an environment variable. VVV sets the value of the environment variable. This is also used to specify DOS settings on DOS and Windows** programs. Each variable/value pair must be separated by a NULL (\0), and the entire string must be terminated by two NULLS (\0\0). For example: DOS DEVICE=value\0env=value\0\0
STARTUPDIR	pathname	Sets the working directory.

Example

```
pszSetupString="PROGTYPE=VDM;"
               "EXENAME=C=\EDIT.EXE;"
               "ASSOCFILTER=*.DOC,*.TXT;"
               "ASSOCTYPE=Plain Text;"
               "SET DOS_DEVICE=1.SYS,2.SYS;"
               "SET PATH=C:\FRED";
```

Class methods

There are currently no class methods defined for the WPProgram Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPProgram class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsCreateDefaultTemplates
- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQuerySettingsPageSize
- wpclsQueryStyle
- wpclsQueryTitle

WPProgramFile

Class definition file: wppgmf.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPDataFile
 WPProgramFile

Description

This is the program file object class. All executable files are of this class. From the Settings notebook pages for objects of this class, it is possible to set up associations to various data file types (files with .TYPE EAs or file extensions). It is also possible to set up a default working

directory and specify parameters to the executable program to be used when it is opened from the shell. An instance of this class can be created as a Workplace object. Instances of this class are created initially by the system for each program file in the system. Each instance has a title corresponding to the file name of the program file it represents and resides in a folder corresponding to the physical directory in which the program file resides.

Menu ID list

The following list shows the WPMENUID menu IDs for this class:

Menu item text	Menu ID
Create another	WPMENUID_CREATEANOTHER
Program	WPMENUID_PROGRAM

Instance methods

The following list shows all the WPProgramFile methods:

- wpAddProgramAssociationPage
- wpAddProgramEnvironmentPage
- wpAddProgramPage
- wpAddProgramSessionPage
- wpQueryAssociationFilter
- wpQueryAssociationType
- wpQueryProgDetails
- wpQueryProgramAssociations
- wpQueryWarnBeforeStart (WPProgramFile)
- wpSetAssociationFilter
- wpSetAssociationType
- wpSetProgDetails
- wpSetProgramAssociations
- wpSetProgIcon
- wpSetWarnBeforeStart (WPProgramFile)

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPProgramFile class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddSettingsPages
- wpConfirmRenameFileWithExt
- wpCopyObject
- wpCreateAnother
- wpDragOver
- wpDrop
- wpFilterMenu
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyMenu
- wpModifyPopupMenu
- wpMoveObject
- wpOpen
- wpQueryNameClashOptions
- wpQueryIcon
- wpRestoreState
- wpSaveState
- wpSetup

wpSetup override by WPProgramFile

The following table shows the keyname-value pairs added by the WPProgramFile class.

KEYNAME	VALUE	DESCRIPTION
ASSOCFILTER	filters	Sets the filename filter for files associated with this program. Multiple

```
                                        filters are
                                        separated by commas.

 ASSOCTYPE       type                   Sets the type of
                                        files associated
                                        with this program.
                                        Multiple types are
                                        separated by commas.
```

Example

```
pszSetupString="PROGTYPE=VDM;"
               "EXENAME=C=\EDIT.EXE;"
               "ASSOCFILTER=*.DOC,*.TXT;"
               "ASSOCTYPE=Plain Text";
```

WPDataFile instance methods

The following list shows all the methods overridden by the WPProgramFile class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpSetAssociatedFileIcon

Class methods

There are currently no class methods defined for the WPProgramFile Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPProgramFile class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQueryInstanceType
- wpclsQuerySettingsPageSize
- wpclsQueryStyle
- wpclsQueryTitle

WPQdr

Class definition file: wpqdr.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient
 WPQdr

Description

This is the queue driver object class. An instance of this class is created by the print object in its Settings view.

Instance methods

There are currently no methods defined for the WPQdr Workplace class object.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDumpSelfInt

WPObject instance methods

The following list shows all the methods overridden by the WPQdr class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddSettingsPages
- wpConfirmDelete
- wpDragOver
- wpDrop
- wpFilterPopupMenu
- wpFree
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyPopupMenu
- wpOpen
- wpQueryIcon
- wpQueryTitle
- wpSetup
- wpUnInitData

Class methods

There are currently no class methods defined for the WPQdr Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPQdr class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIcon
- wpclsQueryStyle
- wpclsQueryTitle

WPRootFolder

Class definition file: wprootf.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPFolder
 WPRootFolder

Description

The WPRootFolder class represents the root folder (or directory) of a file-system device. Instances of this class exist for every file-system device available. Each instance is paired with a WPDisk object, which represents a particular device. An instance of this class can be created as a Workplace object.

Instance methods

There are no methods defined in the WPRootFolder Workplace object class.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows the instance methods overridden by the WPRootFolder class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpFilterMenu
- wpFindViewItem
- wpOpen
- wpQueryDisk

- wpRefresh
- wpRestoreState
- wpSaveState
- wpSetDefaultView
- wpSetTitle
- wpViewObject

WPFileSystem instance methods

The following list shows the instance methods overridden by the WPRootFolder class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddFile1Page
- wpAddFile2Page
- wpAddFile3Page

WPFolder instance methods

The following list shows the instance methods overridden by the WPRootFolder class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpSetFldrSort

Class methods

There are currently no class methods defined for the WPRootFolder Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPRootFolder class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsInitData
- wpclsQueryDefaultView
- wpclsQueryStyle
- wpclsUnInitData

WPRPrinter

Class definition file: wprprint.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPPrinter
 WPRPrinter

Description

This is the remote (network) printer object class. An instance of this class represents a print resource on another computer or server. For proper behavior, a network must be installed. Each instance has a title corresponding to the remote resource and resides on the desktop.

Instance methods

The following list shows all the WPRPrinter methods:

- wpAddNetworkPage
- wpQueryLocalAlias
- wpQueryNetworkId

WPObject instance methods

The following list shows all the methods overridden by the WPRPrinter class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectGeneralPage
- wpAddSettingsPages
- wpCopiedFromTemplate
- wpCopyObject

- wpCreateFromTemplate
- wpDragOver
- wpDrop
- wpFree
- wpFilterPopupMenu
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyPopupMenu
- wpOpen
- wpRestoreState
- wpSaveState
- wpSetup
- wpUnInitData

wpSetup override by WPRPrinter

The following table shows the keyname-value pairs added by the WPRPrinter class.

KEYNAME	VALUE	DESCRIPTION
ICON	filename	The name of the .ICO file to be used as the icon for this object.
NETID	<network>	The full name of the printer resource as it is known to the network. For example: LS:\\DEPTSERV\DEPTPRNT The NETID key will be ignored and FALSE will be returned if this object has already been assigned a NetId.
REFRESHINTERVAL	value	Time interval, in seconds, when the printer object is refreshed.
SHOWJOBS	ALL	All jobs are displayed in the printer object.
	OWN	Only the current user's jobs are displayed in the printer object.
TAKEDEFAULTS	YES	This specifies that the printer object takes the default values when it is created.
	NO	This specifies that the printer object does not take default values when it is created. This will cause the "Access another network printer" dialog to be displayed.

Class methods

There are currently no class methods defined for the WPRPrinter Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPRPrinter class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryIcon
- wpclsQueryIconData
- wpclsQueryStyle
- wpclsQueryTitle

WPSchemePalette

Class definition file: wpscheme.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPPalette
 WPSchemePalette

Description

This is the scheme palette object class. An instance of this class can be created as a Workplace object. An instance of this class is created initially by the system. It has the title "Scheme Palette" and resides in the "System Setup" folder.

Instance methods

There are currently no methods defined for the WPSchemePalette Workplace class object.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPSchemePalette class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpMenuItemHelpSelected
- wpSetup
- wpSetupOnce

wpSetup override by WPSchemePalette

The following are the keyname-value pairs added by the WPSchemePalette class.

KEYNAME	VALUE	DESCRIPTION
AUTOSETUP	YES	This automatically sets the palette values to the original system palette settings. This is the default setting.
	NO	This does not set palette values to the original system palette settings.
SCHEMES	Schemetype	These are the initial schemes for each cell in the scheme palette. The values for each cell are separated by commas. This is equivalent to calling the wpSetupCell instance method. The schemetype value is presented as the scheme name followed by a colon, followed by an application name in the INI file.
XCELLCOUNT	columns	Number of X cells as decimal digits. If not specified, defaults to 4.
YCELLCOUNT	rows	Number of Y cells as decimal digits. If not specified, defaults to 7.

Example

```
pszSetupString="SCHEMES=Marble:PM_Marble_Colors,"
                      "Southwest:PM_Southwest_Colors,"
                      "Khaki:PM_Khaki_Colors";
```

WPPalette instance methods

The following list shows all the methods overridden by the WPSchemePalette class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpDragCell
- wpEditCell
- wpFreeCellData
- wpPaintCell
- wpQueryPaletteHelp
- wpRestoreCellData
- wpSaveCellData
- wpSelectCell
- wpSetupCell

Class methods

There are currently no class methods defined for the WPSchemePalette Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPSchemePalette class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryIconData
- wpclsQueryStyle
- wpclsQueryTitle

WPPalette class methods

The following list shows all the class methods overridden by the WPSchemePalette class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryEditString

WPServer

Class definition file: wpserver.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPFolder
 WPServer

Description

This class is specific to Version 3, or higher, of the OS/2 operating system.

This is the server object class. An instances of this class can be created as a Workplace object. Instances of this class are created for every server in a WPNetgrp folder that is accessible to the user.

Menu ID list

The following list shows the WPMENUID menu IDs for this class:

Menu item text	Menu ID
Access another	WPMENUID_ACCESSNEW
Login...	WPMENUID_LOGIN
Logout...	WPMENUID_LOGOUT
Refresh now	WPMENUID_REFRESH

Instance methods

The following list shows all the WPServer methods.

- wpAddServerPage
- wpQuerySrvrIdentity

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPServer class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddFirstChild
- wpAddObjectGeneralPage
- wpAddObjectGeneralPage2
- wpAddSettingsPages
- wpFilterMenu
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyMenu
- wpModifyMenu
- wpOpen
- wpRestoreState
- wpSaveState
- wpSetup
- wpUnInitData

wpSetup override by WPServer

The following table shows the keyname-value pairs added by the WPServer class.

KEYNAME	VALUE	DESCRIPTION
SRVCOMMENT	comment string	This sets the object's title to the specified comment string. This is equivalent to calling wpSetTitle. Note: The value specified for the TITLE keyword will override this comment.

WPFileSystem instance methods

The following list shows all the methods overridden by the WPServer class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddFile1Page
- wpAddFile2Page
- wpAddFile3Page

WPFolder instance methods

The following list shows all the methods overridden by the WPServer class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddFolderIncludePage
- wpDeleteContents
- wpPopulate

Class methods

There are currently no class methods defined for the WPServer Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPServer class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryIconData
- wpclsQueryIconDataN
- wpclsQuerySettingsPageSize
- wpclsQueryStyle
- wpclsQueryTitle

WPShadow

Class definition file: wpshadow.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPShadow

Description

This is the shadow object class, instances of which provide a persistent link or reference to any other object. This is achieved by storing away the location and identity of the object to which it is linked and then rerouting all requests for help, context menus, and open views to the object to which it is linked.

Instance methods

The following list shows all the WPShadow methods:

- wpQueryShadowedObject
- wpSetLinkToObject
- wpSetShadowTitle

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPShadow class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpConfirmDelete
- wpConfirmObjectTitle
- wpCreateFromTemplate
- wpCreateShadowObject
- wpDraggedOverObject
- wpDragOver
- wpDrop
- wpDroppedOnObject
- wpFilterMenu
- wpFilterPopupMenu
- wpFormatDragItem
- wpInitData
- wpInsertMenuItems
- wpInsertPopupMenuItems
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyMenu
- wpModifyPopupMenu
- wpOpen
- wpPrintObject
- wpQueryDefaultHelp
- wpQueryNameClashOptions
- wpQueryStyle
- wpQueryTitle
- wpRestoreState
- wpSaveState

- wpSetTitle
- wpSetup
- wpUnInitData
- wpViewObject

wpSetup override by WPShadow

The following table shows the keyname-value pairs added by the WPShadow class.

KEYNAME	VALUE	DESCRIPTION
SHADOWID	<name> or file name	This specifies the object for which this object is a shadow. The value for this keyname is an object's id (OBJECTID) or a fully qualified pathname of a directory, program file, or data file.

Class methods

WPObject class methods

There are currently no class methods defined for the WPShadow Workplace class object.

The following list shows all the class methods overridden by the WPShadow class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryIconData
- wpclsQueryStyle
- wpclsQueryTitle

WPSharedDir

Class definition file: wpshdir.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPFolder
 WPSharedDir

Description

This class is specific to Version 3, or higher, of the OS/2 operating system.

This is the shared directory object class. An instances of this class can be created as a Workplace object. Instances of this class are created for every shared directory in a WPServer folder that is accessible to the user.

Menu ID list

The following list shows the WPMENUID menu IDs for this class:

Menu item text	Menu ID
Access another	WPMENUID_ACCESSNEW
Assign drive	WPMENUID_ASSIGN
Login...	WPMENUID_LOGIN
Logout...	WPMENUID_LOGOUT
Refresh now	WPMENUID_REFRESH
Unassign drive	WPMENUID_UNASSIGN

Instance methods

The following list shows all the WPSharedDir methods.

- wpAddResourcePage

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPSharedDir class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectGeneralPage
- wpAddObjectGeneralPage2
- wpAddSettingsPages
- wpFilterMenu
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyMenu
- wpOpen
- wpRestoreState
- wpSaveState

WPFileSystem instance methods

The following list shows all the methods overridden by the WPSharedDir class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddFile1Page
- wpAddFile2Page
- wpAddFile3Page

WPFolder instance methods

The following list shows all the methods overridden by the WPSharedDir class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddFolderIncludePage
- wpPopulate

Class methods

There are currently no class methods defined for the WPSharedDir Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPSharedDir class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQueryIconDataN
- wpclsQuerySettingsPageSize
- wpclsQueryStyle
- wpclsQueryTitle

WPShredder

Class definition file: wpshred.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPShredder

Description

This is the shredder device object class. An instance of this class can be created as a Workplace object. An instance of this class is created initially by the system. It has the title "shredder" and resides on the desktop.

Instance methods

There are currently no methods defined for the WPShadow Workplace class object.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPShredder class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectWindowPage
- wpDragOver
- wpDrop
- wpFilterMenu

Class methods

There are currently no class methods defined for the WPShredder Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPShredder class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQueryStyle
- wpclsQueryTitle

WPSound

Class definition file: wpsound.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPSound

Description

This is the sound object class. An instance of this class can be created as a Workplace object. An instance of this class is created initially by the system. It has the title "Sound" and resides in the "System Setup" folder.

Instance methods

The following list shows all the WPSound methods:

- wpAddSoundSchemePage
- wpAddSoundWarningBeepPage

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPSound class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectWindowPage
- wpAddSettingsPages
- wpFilterMenu

Class methods

There are currently no class methods defined for the WPSound Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPSound class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQueryStyle
- wpclsQueryTitle

WPSpecialNeeds

Class definition file: wpspneed.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPSpecialNeeds

Description

This is the special needs object class. An instance of this class can be created as a Workplace object. An instance of this class is created initially by the system when the keyboard special needs mode is activated via the keyboard object settings. It has the title "Special Needs" and resides on the desktop.

Instance methods

There are currently no methods defined for the WPSpecialNeeds Workplace object class.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPSpecialNeeds class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpFilterMenu

Class methods

There are currently no class methods defined for the WPSpecialNeeds Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPSpecialNeeds class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp

- wpclsQueryIconData
- wpclsQueryTitle

WPSpool

Class definition file: wpspool.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPSpool

Description

This is the spooler object class. An instance of this class is created initially by the system. It has the title "Spooler" and resides in the "System Setup" folder.

Instance methods

The following list shows all the WPSpool class methods:

- wpAddSetPathPage
- wpAddSetPortRedirPage
- wpAddSetPriorityPage
- wpAddSetSyncPage
- wpAddSetViewPage

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDumpSelfInt

WPObject instance methods

The following list shows all the methods overridden by the WPSpool class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddSettingsPages
- wpCopyObject
- wpDragOver
- wpDrop
- wpFilterPopupMenu
- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyPopupMenu
- wpOpen
- wpQueryIcon
- wpQueryTitle
- wpRestoreState
- wpSaveState
- wpUnInitData

Class methods

There are currently no class methods defined for the WPSpool Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPSpool class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIcon
- wpclsQueryIconData
- wpclsQueryStyle

- wpclsQueryTitle

WPStartup

Class definition file: wpstart.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPFolder
 WPStartup

Description

This is the startup folder object class. Any object in the startup folder will be automatically opened every time the system is restarted. Any object in the system that is open at shutdown time will be automatically reopened when the system is restarted. The startup folder is used to automatically open (start) objects that are not necessarily open (running) when the system is shut down. This usually includes things such as a batch file that initializes the network and then terminates. An instance of this class can be created as a Workplace object and is created initially by the system. It has the title "Startup" and resides in the "OS/2 System" folder.

Instance methods

There are currently no methods defined for the WPStartup Workplace object class.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPStartup class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpFree
- wpInitData
- wpRestoreState
- wpSetupOnce

Class methods

There are currently no class methods defined for the WPStartup Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPStartup class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryIconData
- wpclsQueryIconDataN
- wpclsQueryStyle
- wpclsQueryTitle

WPSystem

Class definition file: wpsystem.idl

Class hierarchy
SOMObject

WPObject
WPAbstract
WPSystem

Description

This is the system object class. An instance of this class can be created as a Workplace object and is created initially by the system. It has the title "System" and resides in the "System Setup" folder.

Instance methods

The following list shows all the WPSystem class methods:

- wpAddDMQSDisplayTypePage
- wpAddSysFdrSelfClosePage
- wpAddSysFdrDefViewPage
- wpAddSysFdrMenuPage
- wpAddSystem256ColorPage
- wpAddSystemConfirmationPage
- wpAddSystemInputPage
- wpAddSystemLogoPage
- wpAddSystemPrintScreenPage
- wpAddSystemScreenPage
- wpAddSystemWindowPage
- wpAddTitleConfirmationPage

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPSystem class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectWindowPage
- wpAddSettingsPages
- wpFilterMenu
- wpRestoreState
- wpSaveState

Class methods

There are currently no class methods defined for the WPSystem Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPSystem class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQuerySettingsPageSize
- wpclsQueryStyle
- wpclsQueryTitle

WPAbstract class methods

The following list shows all the class methods overridden by the WPSystem class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQuerySetting
- wpclsSetSetting

wpclsSetSettings override by wpSystem

The defaults can be specified on a wpclsSetSetting invocation to request that the setting be set to its default value. wpclsQuerySetting, however, never returns an xxxx_DEFAULT value.

```
Setting                    Value                  Description
```

Animation	ANIMATION_ON	Enables animation when a window is created or maximized.
	ANIMATION_OFF	Disables animation when a window is created or maximized.
	ANIMATION_DEFAULT	Sets the default window animation setting.
ConfirmCopyMoveCreateShadow	CONFIRM_ON	Displays a confirmation when a copy, move, or create shadow operation occurs.
	CONFIRM_OFF	Does not display a confirmation when a copy, move, or create shadow operation occurs.
	CONFIRM_DEFAULT	Sets the default confirmation display setting.
ConfirmFolderDelete	CONFIRM_ON	Displays a confirmation when a folder is deleted.
	CONFIRM_OFF	Does not display a confirmation when a folder is deleted.
	CONFIRM_DEFAULT	Sets the default confirmation display setting.
ConfirmObjectDelete	CONFIRM_ON	Displays a confirmation when an object is deleted.
	CONFIRM_OFF	Does not display a confirmation when an object is deleted.
	CONFIRM_DEFAULT	Sets the default confirmation display setting.
ConfirmRenameFileExtension	CONFIRM_ON	Displays a confirmation when a file or extension is renamed.
	CONFIRM_OFF	Does not display a confirmation when a file or extension is renamed.
	CONFIRM_DEFAULT	Sets the default confirmation display setting.
CurrentView	CCVIEW_ON	Turns concurrent view mode on.
	CCVIEW_OFF	Turns concurrent view mode off.
	CCVIEW_DEFAULT	Sets the concurrent view mode to the default.
DisplayProgressIndicator	DISPLAY_ON	Displays a progress indicator.
	DISPLAY_OFF	Does not display

		progress indicator.
	DISPLAY_DEFAULT	Sets the default progress indicator display setting.
LogoDisplayTime	LOGO_INDEFINITE	Displays the logo indefinitely.
	LOGO_NONE	Does not display the logo.
	time	Displays the logo for a given period of time. This value must be in the range 0 to 32767 milliseconds.
	LOGO_DEFAULT	Sets the default logo display time.
MinButtonAppearance	BUTTON_HIDE	Sets views of this object to have a hide button as opposed to a minimize button.
	BUTTON_MINIMIZE	Sets views of this object to have a minimize button as opposed to a hide button.
	BUTTON_DEFAULT	Sets the default minimize button appearance setting.
MinButtonBehavior	MINWIN_HIDE	Hides the view of an object when its minimize button is selected.
	MINWIN_VIEWER	Minimizes the view of an object to the minimized window viewer when its minimize button is selected.
	MINWIN_DESKTOP	Minimizes the view of an object to the Desktop when its minimize button is selected.
	MINWIN_DEFAULT	Sets the default minimize button behavior setting.
NameClash	NAMECLASH_PROMPT	Displays a dialog when an object is created, copied or moved to a folder which contains another object with the same name.
	NAMECLASH_RENAME	Renames an object when it is created, copied or moved to a folder which contains another object with the same name.
	NAMECLASH_REPLACE	Replaces the existing object with the new object when an object is created, copied or moved to a folder which contains

		another object with the same name.
	NAMECLASH_DEFAULT	Sets the default setting for handling name clashes.
PrintScreen	PRINTSCREEN_ON	Enables printing when the print screen key is pressed.
	PRINTSCREEN_OFF	Disables printing when the print screen key is pressed.
	PRINTSCREEN_DEFAULT	Sets the default setting when the print screen key is pressed.

WPTemplates

Class definition file: wptemps.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPFolder
 WPTemplates

Description

This is the template folder object class. This class of folder always contains a template instance of every class of object that the user can create that supports the "Create another" action.

A WPTemplates object is a normal folder in all respects other than the following:

- It always contains a template for every class that supports them.
- The last template for each object class cannot be deleted from the folder.

An instance of this class can be created as a Workplace object and is created initially by the system. It has the title, "Templates" and resides on the desktop. A template instance is also created for each data type defined in an application's ASSOCTABLE resource.

Instance methods

There are currently no methods defined for the WPTemplates Workplace object class.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPTemplates class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpSetup

WPFolder instance methods

The following list shows all the methods overridden by the WPTemplates class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpPopulate

Class methods

There are currently no class methods defined for the WPTemplates Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPTemplates class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryIconData
- wpclsQueryIconDataN
- wpclsQueryStyle
- wpclsQueryTitle

WPTransient

Class definition file: wptrans.idl

Class hierarchy
SOMObject
 WPObject
 WPTransient

Description

This is the non-persistent object storage class. The WPTransient class is a storage class with no storage medium. This means that instances of object classes derived from WPTransient do not persist across reboots. This class is available for applications that need to utilize a large amount of workplace functionality (such as context menus and Settings notebooks) in their object class without having to be a file, directory, or a record in the INI file. An instance of this class can be created as a Workplace object. No instances of this class are initially created by the system.

Instance methods

There are currently no methods defined for the WPTransient Workplace object class.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPTransient class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpCopyObject
- wpIdentify
- wpQueryHandle
- wpQueryIcon
- wpSaveImmediate

Class methods

There are currently no class methods defined for the WPTransient Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPTransient class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsObjectFromHandle
- wpclsQueryStyle
- wpclsQueryTitle

WPUrl

Class definition file: wpurl.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPDataFile
 WPUrl

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This is the WPUrl base class. A template for this class is created in the Templates for Internet folder. These objects can be used to represent URIs that will automatically open up a default or specified browser when opened. Instances of this class can also be created by applications, such as web browsers.

When viewed in a WPUrlFolder in the Details view, a WPUrl object displays its *url* string.

Instance methods

There are currently no instance methods defined for the WPUrl Workplace object class.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPUrl class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddSettingsPages
- wpCreateAnother
- wpCreateFromTemplate
- wpObjectReady
- wpOpen
- wpQueryDetailsData
- wpRestoreData
- wpSaveState
- wpSetup
- wpSetupOnce
- wpSwitchTo

wpSetup override by WPUrl

The following table shows the keyname-value pairs added by the WPUrl class.

KEYNAME	VALUE	DESCRIPTION
BROWSER	*name or pathname*	This specifies the executable that will be invoked to display the web page designated in the "Url" field. This field can be either a pathname or a name of a browser in the PATH. EXPLORE.EXE is the default browser for OS/2 Warp. The Java applet viewer (APPLET.EXE) can be used here to view Java applets with the URL object, but be sure to set the "Integrated browser" check box on the Browser page to NO, because APPLET.EXE does not understand the URL-specific browser options.

DEFAULTBROWSER	*name or pathname*	This sets the default value for BROWSER for all URL objects. This value will be placed in the "Path and file name" field on the Browser page of a URL's properties notebook when the Default push button is pressed and when the URL object is first created. Also see the definition for BROWSER.
DEFAULTDISPLAYIMAGES	YES	This sets the default value for DISPLAYIMAGES for all URL objects. This value is used for the "Display images while loading" check box of a URL's properties notebook when the Default push button is pressed or when the URL object is first created. When this keyname is set to YES, the "Display images while loading" check box is checked. Also see the definition for DISPLAYIMAGES.
	NO	The "Display images while loading" check box is unchecked. Also see the definition for DISPLAYIMAGES.
DEFAULTEMAILADDRESS	*address*	This sets the default value for EMAILADDRESS for all URL objects. This value will be placed in the "Electronic mail address" field on the Server page of a URL's properties notebook when the Default push button is pressed and when the URL object is first created. Also see the definition for EMAILADDRESS.
DEFAULTENABLEPROXY	YES	This sets the default value for ENABLEPROXY for all URL objects. The "Proxy gateway" check box on the Server page of a URL's properties notebook is checked when the Default push button is pressed and when the URL object is first created. Also see the definition for ENABLEPROXY.
	NO	The "Proxy gateway" check box is unchecked. Also see the definition for ENABLEPROXY.
DEFAULTENABLESOCKS	YES	This sets the default value for ENABLESOCKS for all URL objects. The "Socks server" check box on the Server page of a URL's properties notebook is checked when the Default push button is pressed or when the URL object is first created. Also see the definition for ENABLESOCKS.
	NO	The "Socks server" check box is unchecked. Also see the definition for ENABLESOCKS.
DEFAULTINTEGRATEDBROWSER	YES	This sets the default value for INTEGRATEDBROWSER for all URL objects. The "Integrated browser" check box on the Browser page of a URL's properties notebook is checked when the Default push button is pressed or when the URL object is first created. Also see the definition for INTEGRATEDBROWSER.
	NO	The "Proxy gateway" check box is unchecked. Also see the definition for

		INTEGRATEDBROWSER.
DEFAULTLOADGRAPHICS	YES	This sets the default value for LOADGRAPHICS for all URL objects. The "Load graphics" check box on the Browser page of a URL's properties notebook is checked when the Default push button is pressed or when the URL object is first created. Also see the definition for LOADGRAPHICS.
	NO	The "Load graphics" check box is unchecked. Also see the definition for LOADGRAPHICS.
DEFAULTNEWSSERVER	*newsserver*	This sets the default value for NEWSERVER for all URL objects. This value will be placed in the "Newsserver" field on the Server page of a URL's properties notebook when the Default push button is pressed and when the URL object is first created. Also see the definition for NEWSERVER.
DEFAULTPALETTEAWARE	YES	This sets the default value for PALETTEAWARE for all URL objects. The "Palette aware" check box on the Web page of a URL's properties notebook is checked when the Default push button is pressed or when the URL object is first created. Also see the definition for PALETTEAWARE.
	NO	The "Palette aware" check box is unchecked. Also see the definition for PALETTEAWARE.
DEFAULTPARAMETERS	*parameters*	This sets the default value for PARAMETERS for all URL objects. This value will be placed in the "Parameters" field on the Browser page of a URL's properties notebook when the Default push button is pressed and when the URL object is first created. Also see the definition for PARAMETERS.
DEFAULTPRESENTATIONMODE	YES	This sets the default value for PRESENTATIONMODE for all URL objects. The "Presentation mode" check box on the Web page of a URL's properties notebook is checked when the Default push button is pressed or when the URL object is first created. Also see the definition for PRESENTATIONMODE.
	NO	The "Presentation mode" check box is unchecked. Also see the definition for PRESENTATIONMODE.
DEFAULTPROXYGATEWAY	*proxy*	This sets the default value for PROXYGATEWAY for all URL objects. This value will be placed in the "Proxy gatewway" field on the Server page of a URL's properties notebook when the Default push button is pressed and when the URL object is first created. Also see the definition for PROXYGATEWAY.
DEFAULTSOCKSSERVER	*socks*	This sets the default value for

		SOCKSERVER for all URL objects. This value will be placed in the "Socks server" field on the Server page of a URL's properties notebook when the Default push button is pressed and when the URL object is first created. Also see the definition for SOCKSERVER.
DEFAULTWORKINGDIR	*directory*	This sets the default value for WORKINGDIR for all URL objects. This value will be placed in the "Working directory" field on the Server page of a URL's properties notebook when the Default push button is pressed and when the URL object is first created. Also see the definition for WORKINGDIR.
DISPLAYIMAGES	YES	This specifies that the browser show the images as they are being constructed on the page while being received from the server. This is the default value for the "Display images while loading" check box on the Web page.
	NO	The "Display images while loading" check box is unchecked. This specifies that the browser show the page when all of the images have been received from the server.
EMAILADDRESS	*address*	This is a required field. This specifies the user's return e-mail address to be used by the browser when responding to other users on mail-to fields of web pages or newsgroup articles. This should be the complete internet e-mail address. For example: marysmith.@ibm.net Leaving this field blank will prevent you from responding to mail-to fields and newsgroup articles.
ENABLEPROXY	YES	This sets the default value for PROXYGATEWAY for all URL objects. If the "Enable proxy?" check box on the Server page is checked. the proxy server will be used.
	NO	If the "Enable proxy?" check box is not checked, the proxy server will not be used.
INTEGRATEDBROWSER	YES	This sets the default value for BROWSER for all URL objects. If the "Integrated browser" check box on the Browser page is checked, this specified that the designated executable has been integrated with the URL object. For example, the IBM WebExplorer browser has been integrated, you would specify YES if you are using EXPLORE.EXE. The Java Applet viewer is not integrated if you are using APPLET.EXE. Other browsers might not be integrated with the URL object. Check the documentation provided with your browser.
	NO	If the "Integrated browser" check box is not checked, you will be unable to use certain browser parameters (for example, Palette Aware, Presentation Mode, and Load

		Graphics) because a non-integrated browser cannot interpret these command-line arguments.
LOADGRAPHICS	YES	This specifies that you can load graphics and images on the web page specified by the URL if the "Load graphics" check box on the Web page is checked.
	NO	If the "Load graphics" check box is not checked, will only be able to view text, but not graphics, on a web page. This will speed up page downloads if you specific not to load graphics.
LOCATOR	*url*	This specifies the Uniform Resource Locator that uniquely identifies each web page designated in the "Uniform Resource Locator (URL)" field on the Web page. The URL is the address for the page on the web. For example: http://www.ibm.net The keywords LOCATOR and URL can be used interchangeably.
NEWSERVER	*news*	This specifies the host name or IP address of the server that handles newsgroups for the user's company or account. The first example is a host name and the second example is an IP address: news.company.com or 128.35.89.2
PALETTEAWARE	YES	This is the default value for BROWSER for all URL objects. This value will be placed in the "Paletteaware" check box on the Web page. This specifies that the OS/2 palette be used to display pages and images.
	NO	The "Paletteaware" check box on the Web page will not be checked. If color problems are detected while displaying web pages with your browser, set this value to NO.
PARAMETERS	*params*	This specifies strings to be included on the command line invocation of (passed to) the executable designated when it is started. If the browser being used permits optional parameters, declare them using this value. This sets the default value for BROWSER for all URL objects. This value will be placed in the "Parameters" field on the Browser page.
PRESENTATIONMODE	YES	This specifies that the browser use the full-screen (nonwindowed) mode. This mode is generally used for presentations. This value is used for the "Presentation mode" check box on the Web page. This is the default value, which specifies that the executable use the OS/2 palette to display its pages and images.
	NO	The "Presentation mode" check box will not be checked. If color problems are detected while displaying web pages with your browser, set this value to NO.

PROXYGATEWAY	*proxy*	This specifies the URL of the server that handles the interface to the web for the user's company or account. It is the *firewall* for insulation of a company from the outside world. A proxy port can also be appended to the end of the proxy string if the proxy server supports it. For example: http://proxy.company.com/ or http://128.35.89.2/ A proxy port can also be appended to the end of the proxy string if the proxy server supports it. For example: http://proxy.company.com:80/ This value will be placed in the "Proxy gateway" field on the Server page. The slash on the end is required syntax. Contact your system administrator for details about using a proxy gateway server on your system.
SOCKSERVER	*socks*	This specifies the socks support that will provide access the web through a firewall, which provides for insulation of a company from the outside world. This value is placed in the "Socks server" field on the Server page. For example: http://socks.company.com/ or http://socks.company.com:80/ The slash on the end is required syntax.
WORKINGDIR	*directory*	This specifies OS/2 working directory for the executable, if required, and sets the default value for BROWSER for all URL objects. This value will be placed in the "Working directory" field on the Browser page. A working directory is required if the specified browser requires DLLs or other files from a directory not specified in the LIBPATH or other environment variables.
URL	*url*	This specifies the Uniform Resource Locator that uniquely identifies each web page designated in the "Uniform Resource Locator (URL)" field on the web page. The URL is the address for a page on the web. For example, an URL might be: http://www.ibm.net The keywords URL and LOCATOR can be used interchangeably.

Class methods

There are currently no class methods defined for the WPUrl Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPUrl class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsCreateDefaultTemplates
- wpclsInitData
- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryDetailsInfo
- wpclsQueryIconData
- wpclsQueryInstanceFilter
- wpclsQueryInstanceType
- wpclsQueryStyle
- wpclsQueryTitle

WPUrlFolder

Class definition file: wpurlfdr.idl

Class hierarchy
SOMObject
 WPObject
 WPFileSystem
 WPDataFile
 WPUrlFolder

Description

This class is specific to Version 4, or higher, of the OS/2 operating system.

This is the WPUrlFolder object class. An instance of this class can be created as a Workplace object. This folder is a subclass of WPFolder and differs from a regular folder in the following ways:

- Has a different default icon, or icons, and title.
- Opens in the Details view by default.
- Shows details of the WPFolder class by default.
- Sorts its contents by default.

A WPUrlFolder object is intended to be a convenient place to store and browse for URLs, but it will accept drops of any WPObject.

Instance methods

There are currently no instance methods defined for the WPUrlFolder Workplace object class.

SOMObject instance methods

The following list shows all the instance methods overridden by the WPObject class. These instance methods are overridden in order to modify the behavior defined by an ancestor class.

- somDefaultInit
- somDestruct

WPObject instance methods

The following list shows all the methods overridden by the WPUrlFolder class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpInitData
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyPopupMenu

Class methods

There are currently no class methods defined for the WPUrlFolder Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPUrlFolder class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsCreateDefaultTemplates
- wpclsInitData
- wpclsQueryDefaultHelp

- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQueryIconDataN
- wpclsQueryStyle
- wpclsQueryTitle

WPWinConfig

Class definition file: wincfg.idl

Class hierarchy
SOMObject
 WPObject
 WPAbstract
 WPWinConfig

Description

This class is specific to Version 3, or higher, of the OS/2 operating system.

This is the WIN-OS/2* configuration object class. This object stores and allows the user to modify the default settings for WIN-OS/2 applications. It initially resides in the system setup folder. An instance of this class can be created as a Workplace object.

Instance methods

The following list shows all the WPWinConfig methods.

- wpAddDDEPage
- wpAddSessionPage
- wpAddSession31Page

WPObject instance methods

The following list shows all the methods overridden by the WPWinConfig class. These methods are overridden in order to modify the behavior defined by an ancestor class.

- wpAddObjectGeneralPage
- wpAddSettingsPages
- wpFilterPopupMenu

Class methods

There are currently no class methods defined for the WPWinConfig Workplace class object.

WPObject class methods

The following list shows all the class methods overridden by the WPWinConfig class. These methods are overridden to modify the behavior defined by an ancestor class.

- wpclsQueryDefaultHelp
- wpclsQueryDefaultView
- wpclsQueryIconData
- wpclsQueryStyle
- wpclsQueryTitle

Workplace Class Methods

This chapter contains an alphabetic listing of the Workplace class methods. Class methods act on class data common to all instances of the class. Metaclasses define all the class methods for a class. Metaclasses are, therefore, the mechanisms for defining class properties as opposed to instance properties. Workplace class methods are prefixed by *wpcls* . See the *Workplace Shell Programming Guide* for guidance information on this subject.

wpclsAddClipboardAssoc

wpclsAddClipboardAssoc - Syntax

This class method adds an association between a clipboard format and a particular class name.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPDataFile     *somSelf;       /*  Pointer to the WPDataFile class object. */
ULONG             aCBFormat;     /*  Atom referring to the clipboard format. */
PSZ               pszClassName;  /*  String containing the name of the class to associate the clipboard format w
ULONG             rc;            /*  Success indicator. */

rc = _wpclsAddClipboardAssoc(somSelf, aCBFormat,
       pszClassName);
```

wpclsAddClipboardAssoc Parameter - somSelf

somSelf (M_WPDataFile *) - input
Pointer to the WPDataFile class object.

wpclsAddClipboardAssoc Parameter - aCBFormat

aCBFormat (ULONG) - input
Atom referring to the clipboard format.

wpclsAddClipboardAssoc Parameter - pszClassName

pszClassName (PSZ) - input
String containing the name of the class to associate the clipboard format with.

wpclsAddClipboardAssoc Return Value - rc

rc (ULONG) - returns
Success indicator.

0	The method class call was successful.
Not 0	The method class call was unsuccessful.

wpclsAddClipboardAssoc - Parameters

somSelf (M_WPDataFile *) - input
Pointer to the WPDataFile class object.

aCBFormat (ULONG) - input
Atom referring to the clipboard format.

pszClassName (PSZ) - input
String containing the name of the class to associate the clipboard format with.

rc (ULONG) - returns
Success indicator.

0	The method class call was successful.
Not 0	The method class call was unsuccessful.

wpclsAddClipboardAssoc - Remarks

This method adds an association between a clipboard format such as "#1" (special name for Plain Text) and a particular class name such as WPDataFile. This is used in the paste-to-folder function to decide which classes support a particular rendering format.

This method is usually called in wpclsInitData when an object class is registered for the first time.

Note: You should block the call to this method with a static variable which forces it to only be called once. Otherwise, it will execute the same code when the object's subclasses are registered.

wpclsAddClipboardAssoc - Usage

This method can be called at any time to associate a private clipboard format to an object class.

wpclsAddClipboardAssoc - How to Override

This method is not generally overridden.

wpclsAddClipboardAssoc - Related Methods

Related Methods

- wpclsInitData

wpclsAddClipboardAssoc - Example Code

This example associates the private clipboard format "MyFormat" with the object class "MyObjectClass".

```
HATOMTBL   hAtomTable = NULLHANDLE;
ATOM       aFormat = 0;

/* Find the system atom table  */
hAtomTable = WinQuerySystemAtomTable();
if (hAtomTable)
{

   /* Add an atom to the system atom table for the clipboard format  */
   aFormat = WinAddAtom(hAtomTable,"MyFormat");
   if (aFormat)
   {

      /* Associate the clipboard format to the object class  */
      _wpclsAddClipBoardAssoc(_WPDataFile,aFormat,"MyObjectClass");
      WinDeleteAtom(hAtomTable,aFormat);
   }
}
```

wpclsAddClipboardAssoc - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpclsCreateDefaultTemplates

wpclsCreateDefaultTemplates - Syntax

This method is called to allow the specified class to create default template instances of its class.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;  /*  Pointer to the WPObject class object. */
WPFolder       *Folder;   /*  Pointer to the folder in which to create the templates. */
BOOL            rc;       /*  Flag indicating whether the class creates the templates. */

rc = _wpclsCreateDefaultTemplates(somSelf,
       Folder);
```

wpclsCreateDefaultTemplates Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsCreateDefaultTemplates Parameter - Folder

Folder (WPFolder *) - input
Pointer to the folder in which to create the templates.

Points to an object of class WPFolder.

wpclsCreateDefaultTemplates Return Value - rc

rc (BOOL) - returns
Flag indicating whether the class creates the templates.

Possible values are described in the following list:

TRUE
The class creates its own templates.
FALSE
The Templates folder creates the templates for the class.

wpclsCreateDefaultTemplates - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

Folder (WPFolder *) - input
Pointer to the folder in which to create the templates.

Points to an object of class WPFolder.

rc (BOOL) - returns
Flag indicating whether the class creates the templates.

Possible values are described in the following list:

TRUE
The class creates its own templates.
FALSE
The Templates folder creates the templates for the class.

wpclsCreateDefaultTemplates - Usage

This method is generally called only by the system when the class is registered. A class is registered by a call to the WinRegisterObjectClass function.

When the system calls this method, "Folder" is a pointer to the Templates folder.

wpclsCreateDefaultTemplates - How to Override

This method should be overridden by classes that need to create default template instances of their class.

wpclsCreateDefaultTemplates - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Glossary

wpclsDecUsage

wpclsDecUsage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This class method is called to decrement the class usage count.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;  /*  Pointer to the WPObject class object. */

wpclsDecUsage(somSelf);
```

wpclsDecUsage Parameter - somSelf

somSelf (M_WPObject *) - input

Pointer to the WPObject class object.

wpclsDecUsage - Return Value

There is no return value for this method.

wpclsDecUsage - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

There is no return value for this method.

wpclsDecUsage - Remarks

The class usage count is used by the system to dynamically load and unload DLLs containing classes.

wpclsDecUsage - How to Override

This method is generally not overridden.

wpclsDecUsage - Related Methods

Related Methods

- wpclsIncUsage

wpclsDecUsage - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpclsEnumClipboardClasses

wpclsEnumClipboardClasses - Syntax

This class method enumerates through all the object classes that a particular rendering mechanism supports.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPDataFile     *somSelf;     /*  Pointer to the WPDataFile class object. */
ULONG             aCBFormat;   /*  Atom referring to the clipboard format. */
ULONG             aPrevious;   /*  The last atom passed in or 0 to start from the beggining. */
ULONG             rv;          /*  An atom referring to a class name, or 0 if finished. */

rv = _wpclsEnumClipboardClasses(somSelf, aCBFormat,
       aPrevious);
```

wpclsEnumClipboardClasses Parameter - somSelf

somSelf (M_WPDataFile *) - input
Pointer to the WPDataFile class object.

wpclsEnumClipboardClasses Parameter - aCBFormat

aCBFormat (ULONG) - input
Atom referring to the clipboard format.

wpclsEnumClipboardClasses Parameter - aPrevious

aPrevious (ULONG) - input
The last atom passed in or 0 to start from the beggining.

wpclsEnumClipboardClasses Return Value - rv

rv (ULONG) - returns
An atom referring to a class name, or 0 if finished.

wpclsEnumClipboardClasses - Parameters

somSelf (M_WPDataFile *) - input
Pointer to the WPDataFile class object.

aCBFormat (ULONG) - input
Atom referring to the clipboard format.

aPrevious (ULONG) - input
The last atom passed in or 0 to start from the beggining.

rv (ULONG) - returns
An atom referring to a class name, or 0 if finished.

wpclsEnumClipboardClasses - Remarks

If the second parameter is zero this method returns the first atom in list. Otherwise, it returns the next class name after the one passed in. If the atom passed in is the last atom in the list the method will return 0.

The class names are returned and passed as atoms. Use WinQueryAtomName to get the real name from the atom.

wpclsEnumClipboardClasses - Usage

This method can be called in a loop to enumerate all of the object classes that can process a specified clipboard format.

wpclsEnumClipboardClasses - How to Override

This method is not generally overridden.

wpclsEnumClipboardClasses - Example Code

This example enumerates all of the object classes that process the "MyFormat" clipboard format.

```
HATOMTBL   hAtomTable = NULLHANDLE;
ATOM       aFormat = 0,aClass = 0;
CHAR       szClassName[CCHMAXSTRING];

/* Find the system atom table  */
hAtomTable = WinQuerySystemAtomTable();
if (hAtomTable)
{

   /* Add an atom to the system atom table for the clipboard format  */
```

```
   aFormat = WinAddAtom(hAtomTable,"MyFormat");
   if (aFormat)
   {

      /* Enumerate the object classes that process the specified
       * clipboard format */
      aClass = _wpclsEnumClipboardClasses(aFormat,(ATOM)0);
      while (aClass)
      {
         if (WinQueryAtomName(hAtomTable,
                              aFormat,
                              szClassName,
                              sizeof(szClassName))
         {
            /***** Process the class whose name was returned in szClassName *****/
         }
         aClass = _wpclsEnumClipboardClasses(aFormat,aClass);
      }

      /* Delete the atom from the system atom table  */
      WinDeleteAtom(hAtomTable,aFormat);
   }
}
```

wpclsEnumClipboardClasses - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Glossary

wpclsFileSysExists

wpclsFileSysExists - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This method returns the object if the specified file or directory already exists in the given folder.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPFileSystem     *somSelf;      /*  Pointer to the WPFileSystem class object. */
SOMAny             *Folder;       /*  Pointer to folder in which to search for file or directory. */
PSZ                 pszFilename;  /*  Name of file or directory on which search is to be performed. */
ULONG               attrFile;     /*  Flag specifying whether name to be searched is a file or directory. */
WPObject            Object *;     /*  Pointer to the object found. */

Object * = _wpclsFileSysExists(somSelf, Folder,
             pszFilename, attrFile);
```

wpclsFileSysExists Parameter - somSelf

somSelf (M_WPFileSystem *) - input
Pointer to the WPFileSystem class object.

wpclsFileSysExists Parameter - Folder

Folder (SOMAny *) - input
Pointer to folder in which to search for file or directory.

wpclsFileSysExists Parameter - pszFilename

pszFilename (PSZ) - input
Name of file or directory on which search is to be performed.

wpclsFileSysExists Parameter - attrFile

attrFile (ULONG) - input
Flag specifying whether name to be searched is a file or directory.

wpclsFileSysExists Return Value - Object *

Object * (WPObject) - returns
Pointer to the object found.

A return value of NULL indicates that the object was not found.

wpclsFileSysExists - Parameters

somSelf (M_WPFileSystem *) - input
Pointer to the WPFileSystem class object.

Folder (SOMAny *) - input
Pointer to folder in which to search for file or directory.

pszFilename (PSZ) - input
Name of file or directory on which search is to be performed.

attrFile (ULONG) - input
Flag specifying whether name to be searched is a file or directory.

Object * (WPObject) - returns
Pointer to the object found.

A return value of NULL indicates that the object was not found.

wpclsFileSysExists - Usage

This method can be called at any time.

wpclsFileSysExists - How to Override

This method is generally not overridden.

wpclsFileSysExists - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Glossary

wpclsFindObjectEnd

wpclsFindObjectEnd - Syntax

This method is called to end the find operation started by a call to the wpclsFindObjectFirst method.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;  /*  Pointer to the WPObject class object. */
HFIND           hfind;    /*  Handle associated with a previous wpclsFindObjectFirst or wpclsFindObjectNext meth
BOOL            rc;       /*  Success indicator. */

rc = _wpclsFindObjectEnd(somSelf, hfind);
```

wpclsFindObjectEnd Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsFindObjectEnd Parameter - hfind

hfind (HFIND) - input
Handle associated with a previous wpclsFindObjectFirst or wpclsFindObjectNext method call.

wpclsFindObjectEnd Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpclsFindObjectEnd - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

hfind (HFIND) - input
Handle associated with a previous wpclsFindObjectFirst or wpclsFindObjectNext method call.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpclsFindObjectEnd - Usage

This method should be called to terminate a find operation that was started by a previous call to the wpclsFindObjectFirst method.

wpclsFindObjectEnd - How to Override

This method is generally not overridden.

wpclsFindObjectEnd - Related Methods

Related Methods

- wpclsFindObjectEnd
- wpclsFindObjectFirst
- wpclsFindObjectNext

wpclsFindObjectEnd - Example Code

This example finds all occurrences of an object type in a folder.

```
 VOID  clsFindEverybody(SOMAny *self)
{
   BOOL      rc;                 /* Return from methods */
   HFIND     hFind       = 0;    /* Handle for repeated finds */
   CLASS     Class[2];           /* List of classes to look for */
   OBJECT    Object[10];         /* Objects returned from search */
   ULONG     ulCount     = 10;   /* Count of objects to find / or found */
   WPFolder  *folder     = 0;    /* Pointer to folder for search */
   ULONG     errorid     = 0;    /* Error returned from previous search */
   CHAR      sz[CCHMAXPATH];     /* Title string to look for */
   PSZ       title = sz;

   Class[0] = _WPFolder;
   Class[1] = NULL;
   /* Get the pointer to the desktop (since that is where we are looking) */

   folder = _wpclsQueryFolder( _WPFolder, "<WP_DESKTOP>", TRUE );

   _wpclsSetError( _WPObject, 0 );      /* Reset the error indicators */
   rc = 0;

         /* Attempt to find the first 10 objects matching our criteria */

   ulCount = 10;
   rc =_wpclsFindObjectFirst(_WPObject,     /* Class self pointer */
                             Class,         /* Pointer to array of class objects */
                             &hFind,        /* Handle for wpclsFindObjectNext */
                             NULL,          /* Find EVERYTHING */
                             folder,        /* Pointer to folder of interest */
                             TRUE,          /* Search all folders in tree */
                             NULL,          /* No extended criteria */
                             Object,        /* Array of object pointers */
                             &ulCount );    /* Number requested and number found */

   errorid = _wpclsQueryError( _WPObject);  /* Get errorid (may still be OK) */

   if ((!rc && (errorid == WPERR_BUFFER_OVERFLOW)) || rc) {

              /***** Process objects found here ******/

   } else {
      somPrintf(" _wpclsFindObjectFirst failed, errorid=%u\n", errorid );
      return;
   } /* endif */

   do {

      _wpclsSetError( _WPObject, 0 );    /* Reset error indicators */
      rc = 0;

             /* Look for next 10 objects */

      rc =_wpclsFindObjectNext(_WPObject,       /* Self */
```

```
                                hFind,          /* Handle from previous Find */
                                Object,         /* Objects found */
                                &ulCount );     /* Number to look for. On
                                                   return, number found */

      errorid = _wpclsQueryError( _WPObject );

      if ((!rc && (errorid == WPERR_BUFFER_OVERFLOW)) || rc) {

              /***** Process objects found here ******/

      } else {
            somPrintf("_wpclsFindObjectNext failed, errorid=%u\n", errorid);
            return;
      } /* endif */

   } while ( errorid == WPERR_BUFFER_OVERFLOW ); /* enddo */

   _wpclsSetError( _WPObject, 0 );    /* Reset error indicators */
   rc = 0;

               /* Indicate find complete */

   rc =_wpclsFindObjectEnd( _WPObject, hFind );

   if (!rc) {
     errorid = _wpclsQueryError( _WPObject );
     somPrintf( "_wpclsFindObjectEnd failed, errorid=%u\n", errorid );
     return;
   }

   return;
}
```

wpclsFindObjectEnd - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpclsFindObjectFirst

wpclsFindObjectFirst - Syntax

This method is called to find workplace objects.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;            /*  Pointer to the WPObject class object. */
PCLASS          pClasslist;         /*  Pointer to an array of class objects. */
PHFIND          phFind;             /*  Address of the handle returned by this method. */
```

```
PSZ             pszTitle;           /*  Pointer to the title specification for objects to be searched. */
WPFolder       *Folder;             /*  Pointer to the folder in which to find objects. */
BOOL            fSubfolders;        /*  Scope indicator. */
PVOID           pExtendedCriteria;  /*  Pointer to the extended search criteria. */
POBJECT         pBuffer;            /*  Pointer to a buffer that contains an array of object pointers. */
PULONG          pCount;             /*  Address of the number of matching entries requested in pBuffer. */
BOOL            rc;                 /*  Success indicator. */

rc = _wpclsFindObjectFirst(somSelf, pClasslist,
       phFind, pszTitle, Folder, fSubfolders,
       pExtendedCriteria, pBuffer, pCount);
```

wpclsFindObjectFirst Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsFindObjectFirst Parameter - pClasslist

pClasslist (PCLASS) - output
Pointer to an array of class objects.

The final element of the array must be NULL.

If this parameter is specified, only objects that are descended from the specified object classes are returned.

wpclsFindObjectFirst Parameter - phFind

phFind (PHFIND) - output
Address of the handle returned by this method.

This handle is used with subsequent calls to wpclsFindObjectNext and wpclsFindObjectEnd.

wpclsFindObjectFirst Parameter - pszTitle

pszTitle (PSZ) - input
Pointer to the title specification for objects to be searched.

This title may include the wildcard characters * and ?.

If this parameter is specified, only objects whose titles match the title specification are returned.

wpclsFindObjectFirst Parameter - Folder

Folder (WPFolder *) - input
Pointer to the folder in which to find objects.

This pointer can be determined by issuing a call to the wpclsQueryFolder method.

wpclsFindObjectFirst Parameter - fSubfolders

fSubfolders (BOOL) - input
Scope indicator.

TRUE	Search Folder and all folders in its tree.
FALSE	Search only the folder specified by Folder.

wpclsFindObjectFirst Parameter - pExtendedCriteria

pExtendedCriteria (PVOID) - input
Pointer to the extended search criteria.

Points to a buffer that contains the class-specific extended search criteria.

If this parameter is specified, only the objects that match the specified extended search criteria are returned.

If this parameter is NULL, objects of all classes are returned.

wpclsFindObjectFirst Parameter - pBuffer

pBuffer (POBJECT) - output
Pointer to a buffer that contains an array of object pointers.

The size of this buffer must be large enough to hold the number of requested entries specified by *pCount*.

wpclsFindObjectFirst Parameter - pCount

pCount (PULONG) - in/out
Address of the number of matching entries requested in *pBuffer*.

On return, this field contains the number of entries placed into *pBuffer*.

wpclsFindObjectFirst Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

If wpclsFindObjectFirst returns FALSE, the wpQueryError method can be called to retrieve the error code. Error codes include the following:

WPERR_BUFFER_OVERFLOW
The *pBuffer* was not large enough to fit all objects matching the specified criteria. The wpclsFindObjectNext method should be called to retrieve the rest of the objects matching the specified criteria.

WPERR_OBJECT_NOT_FOUND
No objects matching the specified criteria were found.

wpclsFindObjectFirst - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

pClasslist (PCLASS) - output
Pointer to an array of class objects.

The final element of the array must be NULL.

If this parameter is specified, only objects that are descended from the specified object classes are returned.

phFind (PHFIND) - output
Address of the handle returned by this method.

This handle is used with subsequent calls to wpclsFindObjectNext and wpclsFindObjectEnd.

pszTitle (PSZ) - input
Pointer to the title specification for objects to be searched.

This title may include the wildcard characters * and ?.

If this parameter is specified, only objects whose titles match the title specification are returned.

Folder (WPFolder *) - input
Pointer to the folder in which to find objects.

This pointer can be determined by issuing a call to the wpclsQueryFolder method.

fSubfolders (BOOL) - input
Scope indicator.

TRUE	Search Folder and all folders in its tree.
FALSE	Search only the folder specified by Folder.

pExtendedCriteria (PVOID) - input
Pointer to the extended search criteria.

Points to a buffer that contains the class-specific extended search criteria.

If this parameter is specified, only the objects that match the specified extended search criteria are returned.

If this parameter is NULL, objects of all classes are returned.

pBuffer (POBJECT) - output
Pointer to a buffer that contains an array of object pointers.

The size of this buffer must be large enough to hold the number of requested entries specified by *pCount*.

pCount (PULONG) - in/out
Address of the number of matching entries requested in *pBuffer*.

On return, this field contains the number of entries placed into *pBuffer*.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

If wpclsFindObjectFirst returns FALSE, the wpQueryError method can be called to retrieve the error code. Error codes include the following:

WPERR_BUFFER_OVERFLOW
The *pBuffer* was not large enough to fit all objects matching the specified criteria. The wpclsFindObjectNext method should be called to retrieve the rest of the objects matching the specified criteria.

WPERR_OBJECT_NOT_FOUND
No objects matching the specified criteria were found.

wpclsFindObjectFirst - Remarks

This method returns object pointers (up to the number requested in *pCount*) for as many objects that match the specifications and that fit in *pBuffer*. On output, *pCount* contains the actual number of object pointers returned. The returned objects are all locked. The wpUnlockObject method for each object must be called when finished using the object.

The wpclsFindObjectNext method uses the find object handle returned by wpclsFindObjectFirst to continue the search started by the wpclsFindObjectFirst request.

The wpclsFindObjectEnd method should be called to terminate the find operation.

wpclsFindObjectFirst - Usage

This method can be called at any time in order to find objects.

wpclsFindObjectFirst - How to Override

This method is generally not overridden.

wpclsFindObjectFirst - Related Methods

Related Methods

- wpclsFindObjectEnd
- wpclsFindObjectNext

wpclsFindObjectFirst - Example Code

This example code finds all folders contained in the Desktop subtree.

```
VOID  clsFindEverybody(SOMAny *self)
{
  BOOL      rc;                 /* Return from methods                */
  HFIND     hFind = 0;          /* Handle for repeated finds          */
  CLASS     Class[2];           /* List of classes to look for        */
  OBJECT    Object[10];         /* Objects returned from search       */
  ULONG     ulCount = 10;       /* Count of objects to find or found  */
  WPFolder  *folder = 0;        /* Pointer to folder for search       */
  ULONG     errorid = 0;        /* Error returned from previous search */
  CHAR      sz[CCHMAXPATH];     /* Title string to look for           */
  PSZ       title = sz;
  Class[0] = _WPFolder;
  Class[1] = NULL;

  /* Get the pointer to the Desktop */
  folder = _wpclsQueryFolder(_WPFolder, "<WP_DESKTOP>", TRUE);

  /* Reset the error indicators */
  _wpclsSetError(_WPObject, 0);

  rc = 0;
  /* Attempt to find the first 10 objects matching the criteria */
  ulCount = 10;
  rc = _wpclsFindObjectFirst(_WPObject,   /* Class self pointer               */
                             Class,       /* Pointer to array of class objects */
                             &hFind,      /* Handle for wpclsFindObjectNext    */
                             NULL,        /* Find EVERYTHING                   */
                             folder,      /* Pointer to folder of interest     */
                             TRUE,        /* Search all folders in tree        */
                             NULL,        /* No extended criteria              */
                             Object,      /* Array of object pointers          */
                             &ulCount);   /* Number requested and number found */

  /* Get the errorid */
  errorid = _wpclsQueryError(_WPObject);

  if ((!rc && (errorid == WPERR_BUFFER_OVERFLOW)) || rc)
  {
    /* Process the objects found here */
  }
  else
  {
    somPrintf(" _wpclsFindObjectFirst failed, errorid=%u\n", errorid);
    return;
  } /* endif */

  do
  {
    /* Reset error indicators */
    _wpclsSetError(_WPObject, 0);

    rc = 0;
    /* Look for the next 10 objects */
    ulCount = 10;
    rc = _wpclsFindObjectNext(_WPObject,  /* Self                        */
                              hFind,      /* Handle from previous Find */
                              Object,     /* Objects found             */
                              &ulCount ); /* Number to look for.       */
                                          /* On return, number found   */
    errorid = _wpclsQueryError(_WPObject);
    if ((!rc && (errorid == WPERR_BUFFER_OVERFLOW)) || rc)
    {
      /* Process all objects found here */
    }
    else
    {
      somPrintf("_wpclsFindObjectNext failed, errorid=%u\n", errorid);
      return;
    } /* endif */
  }
  while (errorid == WPERR_BUFFER_OVERFLOW);

  /* Reset the error indicators */
```

```
  _wpclsSetError(_WPObject, 0);

  rc = 0;
  /* Indicate find complete */
  rc =_wpclsFindObjectEnd(_WPObject, hFind);
  if (!rc)
  {
    errorid = _wpclsQueryError(_WPObject);
    somPrintf("_wpclsFindObjectEnd failed, errorid=%u\n", errorid);
    return;
  }
  return;
}
```

wpclsFindObjectFirst - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpclsFindObjectNext

wpclsFindObjectNext - Syntax

This method is called to find the next set of matching objects.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;  /*  Pointer to the WPObject class object. */
HFIND           hfind;    /*  Handle associated with a previous wpclsFindObjectFirst or wpclsFindObjectNext meth
POBJECT         pBuffer;  /*  Pointer to a buffer that contains an array of object pointers. */
PULONG          pCount;   /*  Address of the number of matching entries requested in pBuffer. */
BOOL            rc;       /*  Success indicator. */

rc = _wpclsFindObjectNext(somSelf, hfind,
       pBuffer, pCount);
```

wpclsFindObjectNext Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsFindObjectNext Parameter - hfind

hfind (HFIND) - input
Handle associated with a previous wpclsFindObjectFirst or wpclsFindObjectNext method call.

wpclsFindObjectNext Parameter - pBuffer

pBuffer (POBJECT) - output
Pointer to a buffer that contains an array of object pointers.

The size of this buffer must be large enough to hold the number of requested entries specified by pulCount.

wpclsFindObjectNext Parameter - pCount

pCount (PULONG) - in/out
Address of the number of matching entries requested in *pBuffer*.

On return, this field contains the number of entries placed into *pBuffer*.

wpclsFindObjectNext Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

FALSE can indicate:
WPERR_OBJECT_NOT_FOUND
WPERR_BUFFER_OVERFLOW

wpclsFindObjectNext - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

hfind (HFIND) - input

Handle associated with a previous wpclsFindObjectFirst or wpclsFindObjectNext method call.

pBuffer (POBJECT) - output
Pointer to a buffer that contains an array of object pointers.

The size of this buffer must be large enough to hold the number of requested entries specified by pulCount.

pCount (PULONG) - in/out
Address of the number of matching entries requested in *pBuffer*.

On return, this field contains the number of entries placed into *pBuffer*.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

FALSE can indicate:
WPERR_OBJECT_NOT_FOUND
WPERR_BUFFER_OVERFLOW

wpclsFindObjectNext - Remarks

wpclsFindObjectNext returns object pointers (up to the number requested in pulCount) for as many objects that match the specifications and that fit in *pBuffer*. On output, pulCount contains the actual number of object pointers returned.

wpclsFindObjectNext uses the find object handle associated with wpclsFindObjectFirst to continue the search started by the wpclsFindObjectFirst request.

If wpclsFindObjectNext returns FALSE, the wpQueryError method can be called to retrieve the error code. Error codes include:

WPERR_OBJECT_NOT_FOUND
No objects matching the specified criteria were found.

WPERR_BUFFER_OVERFLOW
The *pBuffer* buffer was not large enough to fit all objects matching the specified criteria. The wpclsFindObjectNext method should be called to retrieve the rest of the objects matching the specified criteria.

The wpclsFindObjectEnd method should be called to terminate the find operation.

wpclsFindObjectNext - Usage

This method can be called at any time in order to find the next set of matching objects. This method should be called only after a previous call to the wpclsFindObjectFirst method returned an error of WPERR_BUFFER_OVERFLOW.

wpclsFindObjectNext - How to Override

This method is generally not overridden.

wpclsFindObjectNext - Related Methods

Related Methods

- wpclsFindObjectEnd
- wpclsFindObjectFirst
- wpclsFindObjectNext

wpclsFindObjectNext - Example Code

This example finds all occurrences of an object type in a folder.

```
 VOID  clsFindEverybody(SOMAny *self)
{
   BOOL      rc;                 /* Return from methods */
   HFIND     hFind       = 0;    /* Handle for repeated finds */
   CLASS     Class[2];           /* List of classes to look for */
   OBJECT    Object[10];         /* Objects returned from search */
   ULONG     ulCount     = 10;   /* Count of objects to find / or found */
   WPFolder  *folder     = 0;    /* Pointer to folder for search */
   ULONG     errorid     = 0;    /* Error returned from previous search */
   CHAR      sz[CCHMAXPATH];     /* Title string to look for */
   PSZ       title = sz;
   Class[0] = _WPFolder;
   Class[1] = NULL;
   /* Get the pointer to the desktop (since that is where we are looking) */

   folder = _wpclsQueryFolder( _WPFolder, "<WP_DESKTOP>", TRUE );
   _wpclsSetError( _WPObject, 0 );      /* Reset the error indicators */
   rc = 0;
         /* Attempt to find the first 10 objects matching our criteria */

   ulCount = 10;
   rc =_wpclsFindObjectFirst(_WPObject,     /* Class self pointer */
                             Class,         /* Pointer to array of class objects */
                             &hFind,        /* Handle for wpclsFindObjectNext */
                             NULL,          /* Find EVERYTHING */
                             folder,        /* Pointer to folder of interest */
                             TRUE,          /* Search all folders in tree */
                             NULL,          /* No extended criteria */
                             Object,        /* Array of object pointers */
                             &ulCount );    /* Number requested and number found */

   errorid = _wpclsQueryError( _WPObject);  /* Get errorid (may still be OK) */
   if ((!rc && (errorid == WPERR_BUFFER_OVERFLOW)) || rc) {

              /***** Process objects found here ******/
   } else {
      somPrintf(" _wpclsFindObjectFirst failed, errorid=%u\n", errorid );
      return;
   } /* endif */

   do {
      _wpclsSetError( _WPObject, 0 );    /* Reset error indicators */
      rc = 0;
             /* Look for next 10 objects */

      rc =_wpclsFindObjectNext(_WPObject,       /* Self */
                               hFind,           /* Handle from previous Find */
                               Object,          /* Objects found */
                               &ulCount );      /* Number to look for. On
                                                   return, number found */

      errorid = _wpclsQueryError( _WPObject );

      if ((!rc && (errorid == WPERR_BUFFER_OVERFLOW)) || rc) {
              /***** Process objects found here ******/
      } else {
            somPrintf("_wpclsFindObjectNext failed, errorid=%u\n", errorid);
            return;
      } /* endif */

   } while ( errorid == WPERR_BUFFER_OVERFLOW ); /* enddo */

   _wpclsSetError( _WPObject, 0 );    /* Reset error indicators */
   rc = 0;
              /* Indicate find complete */
```

```
   rc =_wpclsFindObjectEnd( _WPObject, hFind );

   if (!rc) {
     errorid = _wpclsQueryError( _WPObject );
     somPrintf( "_wpclsFindObjectEnd failed, errorid=%u\n", errorid );
     return;
   }
   return;
}
```

wpclsFindObjectNext - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpclsFindOneObject

wpclsFindOneObject - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This method is used by find pushbuttons to find an object matching a predefined set of properties.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;        /*  Pointer to the WPObject class object. */
HWND            hwndOwner;      /*  Handle of the owner window for the dialogs. */
PSZ             pszFindParams;  /*  Setup string which defines the properties of an object. */
WPObject       *Object;         /*  Pointer to the found object. */

Object = _wpclsFindOneObject(somSelf, hwndOwner,
           pszFindParams);
```

wpclsFindOneObject Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsFindOneObject Parameter - hwndOwner

hwndOwner (HWND) - input
Handle of the owner window for the dialogs.

wpclsFindOneObject Parameter - pszFindParams

pszFindParams (PSZ) - input
Setup string which defines the properties of an object.

wpclsFindOneObject Return Value - Object

Object (WPObject *) - returns
Pointer to the found object.

wpclsFindOneObject - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

hwndOwner (HWND) - input
Handle of the owner window for the dialogs.

pszFindParams (PSZ) - input
Setup string which defines the properties of an object.

Object (WPObject *) - returns
Pointer to the found object.

wpclsFindOneObject - Remarks

wpSetup override by wpclsFindOneObject

The following table shows the keyname-value pairs for the wpclsFindOneObject method.

KEYNAME	VALUE	DESCRIPTION
CLASSLIST	class-name	This specifies the name of the object class to be searched for. For example, to search for bitmaps you

		would specifiy CLASSLIST=WPBitmap.
DEFAULTCRITERIA	YES	This specifies to use the default file system criteria when doing a search. Hidden files will not be found.
	NO	This specifies to not use the default file system criteria when doing a search.
DIALOG	YES	This specifies that a dialog is to be presented to the user, allowing the user to change the search criteria.
	NO	This specifies that the user cannot change the search criteria. The caller must set up the search criteria exactly as desired.
NAMEFILTER	search-string	This specifies the name of the object to be located. This name may contain wildcard characters. For example, to search for all .BMP files you would specify NAMEFILTER=*.BMP.
STARTFOLDER	path	This specifies the fully-qualified path to the folder from which to start searching for the specified objects. The path name may start with "?:", indicating the specified folder is on the boot drive.
SUBTREESRCH	YES	This specifies that subfolders are to be searched.
	NO	This specifies that subfolders are not to be searched. Only the specified folder is searched.

wpclsFindOneObject - Usage

This method can be called at any time.

wpclsFindOneObject - How to Override

This method is generally not overridden.

wpclsFindOneObject - Related Methods

Related Methods

- wpclsFindObjectEnd
- wpclsFindObjectFirst
- wpclsFindObjectNext

wpclsFindOneObject - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpclsIncUsage

wpclsIncUsage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This class method is called to increment the class usage count.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;  /*  Pointer to the WPObject class object. */

wpclsIncUsage(somSelf);
```

wpclsIncUsage Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsIncUsage - Return Value

There is no return value for this method.

wpclsIncUsage - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

There is no return value for this method.

wpclsIncUsage - Remarks

The class usage count is used by the system to dynamically load and unload DLLs containing classes.

wpclsIncUsage - How to Override

This method should not be overridden.

wpclsIncUsage - Related Methods

Related Methods

- wpclsDecUsage

wpclsIncUsage - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpclsInitData

wpclsInitData - Syntax

This method is called to allow the class object to initialize its instance data.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;  /*  Pointer to the WPObject class object. */

wpclsInitData(somSelf);
```

wpclsInitData Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsInitData - Return Value

There is no return value for this method.

wpclsInitData - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

There is no return value for this method.

wpclsInitData - Remarks

This method is called immediately after the class object is first awakened. When the class object is made dormant, the wpclsUnInitData method is called to give the class object the opportunity to deallocate resources allocated during the processing of wpclsInitData.

wpclsInitData - Usage

This method is generally called only by the system when the class object is awakened. The class object is awakened when the first instance

of this class is either awakened or newly created. It is made dormant again when the last instance of this class is made dormant.

wpclsInitData - How to Override

Any class that has metaclass instance variables should override this method so that those variables are all initially in a known state. It is essential to pass this method onto the parent class object before performing any override processing.

wpclsInitData - Related Methods

Related Methods

- wpclsUnInitData

wpclsInitData - Example Code

This coding example uses wpclsInitData to initialize class data associated with MyDataBase.

```
SOM_Scope void   SOMLINK MyDataBaseM_wpclsInitData(M_MyDataBase *somSelf)
{
    /* M_MyDataBaseData *somThis = M_MyDataBaseGetData(somSelf); */
    M_MyDataBaseMethodDebug("M_MyDataBase","MyDataBaseM_wpclsInitData");

    hModule = _clsQueryModuleHandle( somSelf );

    _hIcon = WinLoadPointer ( HWND_DESKTOP, hModule, ID_WINDOW );

    parent_wpclsInitData(somSelf);
}
```

wpclsInitData - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpclsInsertMultipleObjects

wpclsInsertMultipleObjects - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This method inserts multiple objects into a container at one time.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;        /*  Pointer to the WPObject class object. */
HWND            hwndCnr;        /*  Handle to the container window in which objects are to be inserted. */
PPOINTL         pptlIcon;       /*  Pointer to the initial icon position of the first icon inserted. */
PVOID          *pObjectArray;   /*  Pointer to an array of object pointers. */
PVOID           pRecordParent;  /*  Pointer to the parent record. */
ULONG           NumRecords;     /*  Number of records in pObjectArray to be inserted. */
BOOL            rc;             /*  Success indicator. */

rc = _wpclsInsertMultipleObjects(somSelf,
       hwndCnr, pptlIcon, pObjectArray, pRecordParent,
       NumRecords);
```

wpclsInsertMultipleObjects Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsInsertMultipleObjects Parameter - hwndCnr

hwndCnr (HWND) - input
Handle to the container window in which objects are to be inserted.

wpclsInsertMultipleObjects Parameter - pptlIcon

pptlIcon (PPOINTL) - input
Pointer to the initial icon position of the first icon inserted.

wpclsInsertMultipleObjects Parameter - pObjectArray

pObjectArray (PVOID *) - input
Pointer to an array of object pointers.

The objects in this array are to be inserted into the container.

wpclsInsertMultipleObjects Parameter - pRecordParent

pRecordParent (PVOID) - input
Pointer to the parent record.

This parameter should be set to NULL if the records being inserted will not have a parent or tree view is not supported.

wpclsInsertMultipleObjects Parameter - NumRecords

NumRecords (ULONG) - input
Number of records in *pObjectArray* to be inserted.

This parameter must be greater than 0.

wpclsInsertMultipleObjects Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Errors occurred.

wpclsInsertMultipleObjects - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

hwndCnr (HWND) - input
Handle to the container window in which objects are to be inserted.

pptlIcon (PPOINTL) - input
Pointer to the initial icon position of the first icon inserted.

pObjectArray (PVOID *) - input
Pointer to an array of object pointers.

The objects in this array are to be inserted into the container.

pRecordParent (PVOID) - input
Pointer to the parent record.

This parameter should be set to NULL if the records being inserted will not have a parent or tree view is not supported.

NumRecords (ULONG) - input

Number of records in *pObjectArray* to be inserted.

This parameter must be greater than 0.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.

FALSE
Errors occurred.

wpclsInsertMultipleObjects - Remarks

This method provides rapid insertion of multiple objects into a container at once. wpCnrInsertObject performs the same function as this method, but operates only on one object at a time.

This method adds each object to the in-use list.

wpclsInsertMultipleObjects - Usage

This method can be called at any time to add multiple objects to a container.

wpclsInsertMultipleObjects - How to Override

This method is generally not overridden.

wpclsInsertMultipleObjects - Related Methods

Related Methods

- wpCnrRemoveObject
- wpCnrInsertObject
- wpclsRemoveObjects

wpclsInsertMultipleObjects - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpclsMakeAwake

wpclsMakeAwake - Syntax

This method is called to allow the specified class to awaken an object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;   /*  Pointer to the WPObject class object. */
PSZ             pTitle;    /*  Pointer to the object title. */
ULONG           ulStyle;   /*  Object style flags to set on the object. */
HPOINTER        hptrIcon;  /*  Icon to set on the object. */
POBJDATA        pObjData;  /*  Pointer to the object data. */
WPFolder       *Folder;    /*  Pointer to a Folder object. */
ULONG           ulUser;    /*  Base class-defined value. */
WPObject       *Success;   /*  Success indicator. */

Success = _wpclsMakeAwake(somSelf, pTitle,
            ulStyle, hptrIcon, pObjData, Folder,
            ulUser);
```

wpclsMakeAwake Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsMakeAwake Parameter - pTitle

pTitle (PSZ) - input
Pointer to the object title.

A pointer to a string containing a title to set on the object. If this value is NULL, the class default value is used.

wpclsMakeAwake Parameter - ulStyle

ulStyle (ULONG) - input
Object style flags to set on the object.

If this value is NULL, the current or default value is used.

Predefined object style bits are as follows:

OBJSTYLE_CUSTOMICON
: Destroy the icon when the object goes to sleep.

OBJSTYLE_NOCOPY
: Do not allow copy on the object.

OBJSTYLE_NODELETE
: Do not allow delete on the object.

OBJSTYLE_NODDRAG
: Do not allow drag of this object.

OBJSTYLE_NODDROPON
: Do not allow dropping on this object.

OBJSTYLE_NOLINK
: Do not allow linking on the object.

OBJSTYLE_NOMOVE
: Do not allow moving on the object.

OBJSTYLE_NOPRINT
: Do not allow printting of this object.

OBJSTYLE_NORENAME
: Do not allow renaming of the object.

OBJSTYLE_NOTVISIBLE
: Make this object hidden.

OBJSTYLE_NODSETTINGS
: Do not allow setting of this object.

OBJSTYLE_TEMPLATE
: This object is a template.

wpclsMakeAwake Parameter - hptrIcon

hptrIcon (HPOINTER) - input
: Icon to set on the object.

If this value is NULL, the class default value is used.

wpclsMakeAwake Parameter - pObjData

pObjData (POBJDATA) - input
: Pointer to the object data.

If this value is NULL, the class default value is used.

wpclsMakeAwake Parameter - Folder

Folder (WPFolder *) - input
: Pointer to a Folder object.

A pointer to a Folder object that contains the new object. This pointer can be determined by issuing a call to the wpclsQueryFolder method.

wpclsMakeAwake Parameter - ulUser

ulUser (ULONG) - input
Base class-defined value.

This value is defined by the user and used by the base storage class.

wpclsMakeAwake Return Value - Success

Success (WPObject *) - returns
Success indicator.

NULL	Error occurred
Other	Pointer to the awakened object.

wpclsMakeAwake - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

pTitle (PSZ) - input
Pointer to the object title.

A pointer to a string containing a title to set on the object. If this value is NULL, the class default value is used.

ulStyle (ULONG) - input
Object style flags to set on the object.

If this value is NULL, the current or default value is used.

Predefined object style bits are as follows:

OBJSTYLE_CUSTOMICON
Destroy the icon when the object goes to sleep.
OBJSTYLE_NOCOPY
Do not allow copy on the object.
OBJSTYLE_NODELETE
Do not allow delete on the object.
OBJSTYLE_NODDRAG
Do not allow drag of this object.
OBJSTYLE_NODDROPON
Do not allow dropping on this object.
OBJSTYLE_NOLINK
Do not allow linking on the object.
OBJSTYLE_NOMOVE
Do not allow moving on the object.
OBJSTYLE_NOPRINT
Do not allow printting of this object.
OBJSTYLE_NORENAME
Do not allow renaming of the object.
OBJSTYLE_NOTVISIBLE
Make this object hidden.
OBJSTYLE_NODSETTINGS
Do not allow setting of this object.
OBJSTYLE_TEMPLATE
This object is a template.

hptrIcon (HPOINTER) - input
Icon to set on the object.

If this value is NULL, the class default value is used.

pObjData (POBJDATA) - input
Pointer to the object data.

If this value is NULL, the class default value is used.

Folder (WPFolder *) - input
Pointer to a Folder object.

A pointer to a Folder object that contains the new object. This pointer can be determined by issuing a call to the wpclsQueryFolder method.

ulUser (ULONG) - input
Base class-defined value.

This value is defined by the user and used by the base storage class.

Success (WPObject *) - returns
Success indicator.

NULL	Error occurred
Other	Pointer to the awakened object.

wpclsMakeAwake - Remarks

An object is made awake when it is created in an open folder or when a folder containing the object is opened. An object awakened by the *wpclsMakeAwake* class method is automatically locked. The wpUnlockObject method can be called to allow the object to return to its dormant state.

wpclsMakeAwake - Usage

This method is generally called only by base storage classes in order to awaken an object from the dormant state.

wpclsMakeAwake - How to Override

This method is generally not overridden.

wpclsMakeAwake - Related Methods

Related Methods

- wpclsNew
- wpInitData
- wpUnlockObject

wpclsMakeAwake - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpclsNew

wpclsNew - Syntax

This method creates a new instance of a given class of object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;      /*  Pointer to the WPObject class object. */
PSZ             pszTitle;     /*  Pointer to the object title. */
PSZ             pszSetupEnv;  /*  Environment setup string. */
WPFolder       *Folder;       /*  Pointer to the folder object in which to place the new object. */
BOOL            fLock;        /*  Lock object flag. */
WPObject       *wpclsNew;     /*  Pointer to the new object. */

wpclsNew = _wpclsNew(somSelf, pszTitle, pszSetupEnv,
             Folder, fLock);
```

wpclsNew Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsNew Parameter - pszTitle

pszTitle (PSZ) - input
Pointer to the object title.

A pointer to a null-terminated string which contains the initial title of the object as it is to appear when displayed on the user interface underneath an icon or on the title bar of an open object.

wpclsNew Parameter - pszSetupEnv

pszSetupEnv (PSZ) - input
Environment setup string.

A pointer to a zero-terminated string which contains the environment setup string of the new object. The *pszSetupEnv* string is extracted when the wpSetup method is called.

Refer to wpSetup for the description of this field for the class of the object being created.

wpclsNew Parameter - Folder

Folder (WPFolder *) - input
Pointer to the folder object in which to place the new object.

This pointer can be determined by issuing a call to the wpclsQueryFolder method.

wpclsNew Parameter - fLock

fLock (BOOL) - input
Lock object flag.

Possible values are described in the following list:

TRUE
The new flag remains awake until the caller issues the wpUnlockObject method on it.

FALSE
The newly created object is made dormant whenever the object and the folder containing the object are closed.

wpclsNew Return Value - wpclsNew

wpclsNew (WPObject *) - returns
Pointer to the new object.

A return value of NULL indicates that an error occurred.

wpclsNew - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

pszTitle (PSZ) - input
Pointer to the object title.

A pointer to a null-terminated string which contains the initial title of the object as it is to appear when displayed on the user interface

underneath an icon or on the title bar of an open object.

pszSetupEnv (PSZ) - input
Environment setup string.

A pointer to a zero-terminated string which contains the environment setup string of the new object. The *pszSetupEnv* string is extracted when the wpSetup method is called.

Refer to wpSetup for the description of this field for the class of the object being created.

Folder (WPFolder *) - input
Pointer to the folder object in which to place the new object.

This pointer can be determined by issuing a call to the wpclsQueryFolder method.

fLock (BOOL) - input
Lock object flag.

Possible values are described in the following list:

TRUE
The new flag remains awake until the caller issues the wpUnlockObject method on it.

FALSE
The newly created object is made dormant whenever the object and the folder containing the object are closed.

wpclsNew (WPObject *) - returns
Pointer to the new object.

A return value of NULL indicates that an error occurred.

wpclsNew - Remarks

This method is a modified version of somNew that takes arguments. These arguments provide a way to create a new object with a defined state.

wpclsNew - Usage

This method can be called at any time to create a new workplace object. The created object is a persistent instance of the class specified by WPClass.

wpclsNew - How to Override

This method can be overridden only in a metaclass.

wpclsNew - Related Methods

Related Methods

- wpFree
- wpSetup
- wpUnlockObject
- wpclsQueryFolder

wpclsNew - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpclsObjectFromHandle

wpclsObjectFromHandle - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This method extracts the object pointer for an instance represented by the given handle.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;  /*  Pointer to the WPObject class object. */
HOBJECT         hObject;  /*  Persistent object handle. */
WPObject       *pObject;  /*  Pointer to the object instance. */

pObject = _wpclsObjectFromHandle(somSelf,
            hObject);
```

wpclsObjectFromHandle Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsObjectFromHandle Parameter - hObject

hObject (HOBJECT) - input
Persistent object handle.

wpclsObjectFromHandle Return Value - pObject

pObject (WPObject *) - returns
Pointer to the object instance.

wpclsObjectFromHandle - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

hObject (HOBJECT) - input
Persistent object handle.

pObject (WPObject *) - returns
Pointer to the object instance.

wpclsObjectFromHandle - Remarks

Each base class must store the relationship between the object handles and the folders they reside in.

Using HOBJECT for .INI files or files in which an application uses a rename/save/delete sequence is not supported.

wpclsObjectFromHandle - Usage

This method can be called at any time to find the object pointer from a given handle.

wpclsObjectFromHandle - How to Override

This method is generally not overridden.

wpclsObjectFromHandle - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Glossary

wpclsQueryActiveDesktop

wpclsQueryActiveDesktop - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This method returns a pointer to the active WPDesktop object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPDesktop     *somSelf;    /*  Pointer to the WPDesktop class object. */
WPDesktop       *WPDesktop;  /*  Pointer to the active desktop object. */

WPDesktop = _wpclsQueryActiveDesktop(somSelf);
```

wpclsQueryActiveDesktop Parameter - somSelf

somSelf (M_WPDesktop *) - input
Pointer to the WPDesktop class object.

wpclsQueryActiveDesktop Return Value - WPDesktop

WPDesktop (WPDesktop *) - returns
Pointer to the active desktop object.

wpclsQueryActiveDesktop - Parameters

somSelf (M_WPDesktop *) - input
Pointer to the WPDesktop class object.

WPDesktop (WPDesktop *) - returns
Pointer to the active desktop object.

wpclsQueryActiveDesktop - Remarks

Multiple desktop objects can be awake, but only one can be active at a time.

wpclsQueryActiveDesktop - Usage

This method can be called at any time.

wpclsQueryActiveDesktop - How to Override

This method is generally not overridden.

wpclsQueryActiveDesktop - Related Methods

Related Methods

- wpclsQueryActiveDesktopHWND

wpclsQueryActiveDesktop - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpclsQueryActiveDesktopHWND

wpclsQueryActiveDesktopHWND - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This method returns the handle of the active WPDesktop object.

```
#define INCL_WINWORKPLACE
#include <os2.h>
```

```
M_WPDesktop     *somSelf;      /*  Pointer to the WPDesktop class object. */
HWND             hwndDesktop;  /*  Handle of the active desktop object' frame window. */

hwndDesktop = _wpclsQueryActiveDesktopHWND(
               somSelf);
```

wpclsQueryActiveDesktopHWND Parameter - somSelf

somSelf (M_WPDesktop *) - input
Pointer to the WPDesktop class object.

wpclsQueryActiveDesktopHWND Return Value - hwndDesktop

hwndDesktop (HWND) - returns
Handle of the active desktop object' frame window.

wpclsQueryActiveDesktopHWND - Parameters

somSelf (M_WPDesktop *) - input
Pointer to the WPDesktop class object.

hwndDesktop (HWND) - returns
Handle of the active desktop object' frame window.

wpclsQueryActiveDesktopHWND - Usage

This method can be called at any time.

wpclsQueryActiveDesktopHWND - How to Override

This method is generally not overridden.

wpclsQueryActiveDesktopHWND - Related Methods

Related Methods

- wpclsQueryActiveDesktop

wpclsQueryActiveDesktopHWND - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpclsQueryAwakeObject

wpclsQueryAwakeObject - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This class method determines if the given object is already awake.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPFileSystem     *somSelf;       /*  Pointer to the WPFileSystem class object. */
PSZ                 pszInputPath;  /*  Fully-qualified input path. */
WPObject            Object;        /*  The awake object. */

Object = _wpclsQueryAwakeObject(somSelf, pszInputPath);
```

wpclsQueryAwakeObject Parameter - somSelf

somSelf (M_WPFileSystem *) - input
Pointer to the WPFileSystem class object.

wpclsQueryAwakeObject Parameter - pszInputPath

pszInputPath (PSZ) - input
Fully-qualified input path.

Note: This parameter must be a valid path. The system does not call the file system to check whether the file exists.

wpclsQueryAwakeObject Return Value - Object

Object (WPObject) - returns
The awake object.

A return value of NULL indicates that the object is not already awake.

wpclsQueryAwakeObject - Parameters

somSelf (M_WPFileSystem *) - input
Pointer to the WPFileSystem class object.

pszInputPath (PSZ) - input
Fully-qualified input path.

Note: This parameter must be a valid path. The system does not call the file system to check whether the file exists.

Object (WPObject) - returns
The awake object.

A return value of NULL indicates that the object is not already awake.

wpclsQueryAwakeObject - Remarks

This methods requires that for an object to be awake, all path coomponents prior to the object are also awake. If any path component is not found awake, then the routine assumes that the object for that passed file cannot be awake either.

wpclsQueryAwakeObject - Usage

This method can be called at any time.

wpclsQueryAwakeObject - How to Override

This method is generally not overridden.

wpclsQueryAwakeObject - Related Methods

Related Methods

- wpclsMakeAwake

wpclsQueryAwakeObject - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpclsQueryButtonAppearance

wpclsQueryButtonAppearance - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This class method returns the system button appearance.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;       /*  Pointer to the WPObject class object. */
ULONG           ulButtonType;  /*  Flag indicating the system default button appearance. */

ulButtonType = _wpclsQueryButtonAppearance(
                somSelf);
```

wpclsQueryButtonAppearance Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsQueryButtonAppearance Return Value - ulButtonType

ulButtonType (ULONG) - returns
Flag indicating the system default button appearance.

Possible values are described in the following list:

HIDEBUTTON

MINBUTTON

wpclsQueryButtonAppearance - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

ulButtonType (ULONG) - returns
Flag indicating the system default button appearance.

Possible values are described in the following list:

HIDEBUTTON

MINBUTTON

wpclsQueryButtonAppearance - Usage

This method can be called at any time to query the default button appearance.

wpclsQueryButtonAppearance - How to Override

This method is generally not overridden.

wpclsQueryButtonAppearance - Related Methods

Related Methods

- wpQueryButtonAppearance
- wpSetButtonAppearance

wpclsQueryButtonAppearance - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage

Related Methods
Glossary

wpclsQueryDefaultHelp

wpclsQueryDefaultHelp - Syntax

This method is called to allow the class object to specify its default help panel for its instances.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;      /*  Pointer to the WPObject class object. */
PULONG          HelpPanelId;  /*  Pointer to the help panel which provides help for this class. */
PSZ             HelpLibrary;  /*  Pointer to the buffer in which to place the name of the Help library. */
ULONG           rc;           /*  Success indicator. */

rc = _wpclsQueryDefaultHelp(somSelf, HelpPanelId,
       HelpLibrary);
```

wpclsQueryDefaultHelp Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsQueryDefaultHelp Parameter - HelpPanelId

HelpPanelId (PULONG) - output
Pointer to the help panel which provides help for this class.

wpclsQueryDefaultHelp Parameter - HelpLibrary

HelpLibrary (PSZ) - output
Pointer to the buffer in which to place the name of the Help library.

This buffer should be at least the length of CCHMAXPATH bytes.

wpclsQueryDefaultHelp Return Value - rc

rc (ULONG) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpclsQueryDefaultHelp - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

HelpPanelId (PULONG) - output
Pointer to the help panel which provides help for this class.

HelpLibrary (PSZ) - output
Pointer to the buffer in which to place the name of the Help library.

This buffer should be at least the length of CCHMAXPATH bytes.

rc (ULONG) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpclsQueryDefaultHelp - Remarks

This class method is called during the default processing of wpQueryDefaultHelp.

wpclsQueryDefaultHelp - Usage

This method can be called at any time in order to determine the default help panel for this object class.

wpclsQueryDefaultHelp - How to Override

The default WPObject class does not process this method other than returning FALSE.

wpclsQueryDefaultHelp - Related Methods

Related Methods

- wpDisplayHelp
- wpMenuItemHelpSelected
- wpQueryDefaultHelp
- wpSetDefaultHelp
- wpclsQueryDefaultHelp

wpclsQueryDefaultHelp - Example Code

This example provides general help for the Spread (sheet) instance.

```
SOM_Scope BOOL   SOMLINK SpreadM_wpclsQueryDefaultHelp(M_Spread *somSelf,
                 PULONG pHelpPanelId,
                 PSZ pszHelpLibrary)
{
    /* M_SpreadData *somThis = M_SpreadGetData(somSelf); */
    M_SpreadMethodDebug("M_Spread","SpreadM_wpclsQueryDefaultHelp");

    if (pHelpPanelId)                          /* set default help panel id */
       *pHelpPanelId = PANEL_MAIN;

    if (pszHelpLibrary)                        /* copy help filename */
       strcpy( pszHelpLibrary, "spdsheet.hlp");

    return( TRUE );    /* Help exists */
}
```

wpclsQueryDefaultHelp - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpclsQueryDefaultView

wpclsQueryDefaultView - Syntax

This method is called to allow the class object to specify the default open view for its instance.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;  /*  Pointer to the WPObject class object. */
```

```
ULONG           ulView;   /*  Default open view. */

ulView = wpclsQueryDefaultView(somSelf);
```

wpclsQueryDefaultView Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsQueryDefaultView Return Value - ulView

ulView (ULONG) - returns
Default open view.

OPEN_CONTENTS
Open contents view.
OPEN_DEFAULT
Open default view (same as double-click).
OPEN_DETAILS
Open details view.
OPEN_HELP
Display HelpPanel.
OPEN_RUNNING
Execute object.
OPEN_SETTINGS
Open Settings notebook.
OPEN_TREE
Open tree view.
OPEN_UNKNOWN
Unknown view.
OPEN_USER
Class-specific views have a greater value than this.

wpclsQueryDefaultView - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

ulView (ULONG) - returns
Default open view.

OPEN_CONTENTS
Open contents view.
OPEN_DEFAULT
Open default view (same as double-click).
OPEN_DETAILS
Open details view.
OPEN_HELP
Display HelpPanel.
OPEN_RUNNING
Execute object.
OPEN_SETTINGS
Open Settings notebook.
OPEN_TREE

Open tree view.

OPEN_UNKNOWN
Unknown view.

OPEN_USER
Class-specific views have a greater value than this.

wpclsQueryDefaultView - Usage

This method can be called at any time in order to query the default open view for instances of this class.

wpclsQueryDefaultView - How to Override

All classes should override this method, so that new objects in their class will always have a sensible default view (device objects typically have a default view of OPEN_SETTINGS). The default view is used for both the conditional Open cascade menu and double-clicking on the object.

wpclsQueryDefaultView - Related Methods

Related Methods

- wpQueryDefaultView
- wpclsQueryDefaultView
- wpSetDefaultView

wpclsQueryDefaultView - Example Code

This example demonstrates how to change the default open view of the instances of this class by overriding this method.

```
SOM_Scope ULONG   SOMLINK MyObjectM_wpclsQueryDefaultView(M_MyObject *somSelf)
{
    /* M_MyObjectData *somThis = M_MyObjectGetData(somSelf); */
    M_MyObjectMethodDebug("M_MyObject","MtObjectM_wpclsQueryDefaultView");

    /*  Make the default open view the ICON view */

    return ( OPEN_ICON );
}
```

wpclsQueryDefaultView - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code

Related Methods
Glossary

wpclsQueryDetails

wpclsQueryDetails - Syntax

This method is called to allow the class object to specify the default details view items for its instances.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject      *somSelf;  /*  Pointer to the WPObject class object. */
PCLASSDETAILS    pReturn;  /*  Success indicator. */

pReturn = _wpclsQueryDetails(somSelf);
```

wpclsQueryDetails Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsQueryDetails Return Value - pReturn

pReturn (PCLASSDETAILS) - returns
Success indicator.

NULL	Error occurred
Other	Pointer to details information.

wpclsQueryDetails - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

pReturn (PCLASSDETAILS) - returns
Success indicator.

NULL	Error occurred
Other	Pointer to details information.

wpclsQueryDetails - Related Methods

Related Methods

- wpQueryDetailsData
- wpclsQueryDetails
- wpclsQueryDetailsInfo

wpclsQueryDetails - Example Code

This example shows an override of the method to invoke 'DisplayMethodInfo' in the WPSTUTOR sample program.

```
SOM_Scope PCLASSDETAILS   SOMLINK wpstutM_wpclsQueryDetails(M_WPSTutorial *somSelf)
{
    /* M_WPSTutorialData *somThis = M_WPSTutorialGetData(somSelf); */
    M_WPSTutorialMethodDebug("M_WPSTutorial","wpstutM_wpclsQueryDetails");

    DisplayMethodInfo(IDS_WPCLSQUERYDETAILS);

    return (parent_wpclsQueryDetails(somSelf));
}
```

wpclsQueryDetails - Topics

Select an item:
Syntax
Parameters
Returns
Example Code
Related Methods
Glossary

wpclsQueryDetailsInfo

wpclsQueryDetailsInfo - Syntax

This method is called to allow the class object to specify its details to be used for its instances.

```
#define INCL_WINWORKPLACE
#include <os2.h>
```

```
M_WPObject        *somSelf;          /*  Pointer to the WPObject class object. */
PCLASSFIELDINFO    pClassFieldInfo;  /*  Pointer to details information. */
PULONG             pSize;            /*  Total number of bytes of details data. */
ULONG              ulReturn;         /*  Sum of the number of detail columns for the object. */

ulReturn = _wpclsQueryDetailsInfo(somSelf,
             pClassFieldInfo, pSize);
```

wpclsQueryDetailsInfo Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsQueryDetailsInfo Parameter - pClassFieldInfo

pClassFieldInfo (PCLASSFIELDINFO) - in/out
Pointer to details information.

wpclsQueryDetailsInfo Parameter - pSize

pSize (PULONG) - in/out
Total number of bytes of details data.

This total includes the details added by this class and ancestor classes.

wpclsQueryDetailsInfo Return Value - ulReturn

ulReturn (ULONG) - returns
Sum of the number of detail columns for the object.

This sum includes details added by this class and ancestor classes.

wpclsQueryDetailsInfo - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

pClassFieldInfo (PCLASSFIELDINFO) - in/out
Pointer to details information.

pSize (PULONG) - in/out
Total number of bytes of details data.

This total includes the details added by this class and ancestor classes.

ulReturn (ULONG) - returns
Sum of the number of detail columns for the object.

This sum includes details added by this class and ancestor classes.

wpclsQueryDetailsInfo - Remarks

All objects which have information to display in details view must override this method.

The two possible queries are:

Query 1

A request for the CLASSFIELDINFO linked the list segment associated with an object. This information is needed just prior to changing the view of a container control to details view.

If *pClassFieldInfo* is NULL (no subclasses have appended details data), the number of columns of details information for this class and all its superclasses is returned. If *pClassFieldInfo* is NON-NULL, the number of columns of details informtion is returned and **pClassFieldInfo* points to the head of a linked list of CLASSFIELDINFO structures to which a linked list of CLASSFIELDINFO structures describing the details fields of objects of this subclass should be appended.

For example,on input **pClassFieldInfo* :

```
CLASSFIELDINFO_1_grandparent_
CLASSFIELDINFO_2_grandparent
 CLASSFIELDINFO_1_parent
```

on output, **pClassFieldInfo* :

```
CLASSFIELDINFO_1_grandparent_
CLASSFIELDINFO_2_grandparent
 CLASSFIELDINFO_1_parent
CLASSFIELDINFO_1_self
```

The correct way to handle this request is to do the following:

- Call the parent method. The parent method returns the number of columns the ancestors have contributed.
- Walk the chain of CLASSFIELDINFO structures to the end and append the chain of CLASSFIELDINFO structures for the current class (using the *pNextFieldInfo* element).
- If **pClassFieldInfo* is zero (indicating no parent columns), assign **pClassFieldInfo* = beginning of CLASSFIELDINFO chain for this subclass.
- Return the sum of the details columns of the parent and the current subclass.

Note the following differences between the CLASSFIELDINFO and FIELDINFO structures:

- The *offFieldData* and *ulLenFieldData* fields are required so that appropriate offsets for the data can be computed. The application is responsible for providing *offFieldData* and *ulLenFieldData* fields.
- The *ulReserved* field should be left NULL.
- Owner draw, comparison, and sort functions may be specified for the field.

For example, if an object has three fields:

```
typedef struct _SAMPLE_DETAIL_DATA {
  CDATE   cdate;
```

```
  CTIME   ctime;
  PSZ     psz;
}  SAMPLE_DETAIL_DATA;

classfieldinfo[0].offFieldData   = FIELDOFFSET(SAMPLE,cdate);
classfieldinfo[0].ulLenFieldData = FIELDOFFSET(SAMPLE,ctime)
                                       FIELDOFFSET(SAMPLE,cdate);

classfieldinfo[1].offFieldData   = FIELDOFFSET(SAMPLE,ctime);
classfieldinfo[1].ulLenFieldData = FIELDOFFSET(SAMPLE,psz) -
                                       FIELDOFFSET(SAMPLE,ctime);

classfieldinfo[2].offFieldData   = FIELDOFFSET(SAMPLE,psz);
classfieldinfo[2].ulLenFieldData = sizeof(SAMPLE) -
                                       FIELDOFFSET(SAMPLE,psz);
```

Note that **pClassFieldInfo* must be modified to point to the beginning of the CLASSFIELDINFO linked list only if **pClassFieldInfo* is 0.

The application must return the sum of the details columns of the parent and itself:

```
return(n_cols + n_parent_cols);
```

Note also that the fields MUST be set up sequentially. classfieldinfo[0] must point to data at an offset of 0. classfieldinfo[n] must point to data adjacent, and directly following that described by classfieldinfo[-1].

Query 2

The number of bytes of details data associated with an object. This information is needed prior to allocating memory for a container control insert record.

If *pSize* is NON-NULL, the override should adjust **pSize* by the number of bytes which must be added to the end of a MINIRECORDCORE structure to hold the details information for objects of this class.

Example:

```
*pSize += bytes_of_details_data;
```

In the case of the above example,

```
*pSize += sizeof(SAMPLE_DETAIL_DATA);
```

Note: All class field pointers returned should be pointers to static data areas.

wpclsQueryDetailsInfo - Usage

This method is generally called only by the system.

wpclsQueryDetailsInfo - How to Override

This method should be overridden by classes that introduce class-specific details to be displayed in details view. The parent method must always be called before appending the request information.

wpclsQueryDetailsInfo - Related Methods

Related Methods

- wpQueryDetailsData
- wpclsQueryDetails
- wpclsQueryDetailsInfo

wpclsQueryDetailsInfo - Example Code

This example adds details data for the Car object by appending FIELDINFO structures to *ppClassFieldInfo.

```
SOM_Scope ULONG SOMLINK carM_wpclsQueryDetailsInfo(M_Car *somSelf,
                PCLASSFIELDINFO *ppClassFieldInfo,
                PULONG pSize)
{
 ULONG           cParentCol;
 PCLASSFIELDINFO pCFI;
 ULONG           i;

    /* M_CarData *somThis = M_CarGetData(somSelf); */
    M_CarMethodDebug("M_Car","carM_wpclsQueryDetailsInfo");

   /*  Always call the parent method first to retrieve the number of
       details columns and any data already defined in the details buffer. */

   cParentCol = parent_wpclsQueryDetailsInfo(somSelf, ppClassFieldInfo, pSize);

        /* If details columns exist, add the size of ours to it */

   if (pSize)
      *pSize += sizeof(CARDETAILS);

   /*   If the request was for the chained fieldinfo structures
    *   (ppClassFieldInfo is non-NULL), link them in
    *
    *   eventually the chain will look like
    *
    *   Grandad - Dad - Me - Kid - Grandkid
    *
    *   I will be getting the pointer to the beginning of the chain
    *
    *   If the beginning of the chain is 0, I will assign the address
    *   of my first CLASSFIELDINFO structure to *ppClassFieldInfo.
    *   Otherwise *pp points to the first column description in the
    *   chain.  We need to walk the chain and link our CLASSFIELDINFO
    *   structures at the end.
    */

   if (ppClassFieldInfo) {

      if (*ppClassFieldInfo) {
         pCFI = *ppClassFieldInfo;
         for (i=0;i<cParentColumns;i++)
            pCFI = (pCFI->pNextFieldInfo) ? pCFI->pNextFieldInfo : pCFI;

         pCFI->pNextFieldInfo = fieldinfo;
      } else {
         *ppClassFieldInfo = fieldinfo;
      } /* endif */

   } /* endif */

   return ((ULONG) (cParentColumns + NUM_CAR_FIELDS));
}
```

wpclsQueryDetailsInfo - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpclsQueryEditString

wpclsQueryEditString - Syntax

This method is called to allow the class object to specify the text to be used in the edit pushbutton of the palette object's open view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPPalette     *somSelf;   /*  Pointer to the WPPalette class object. */
PSZ              ppointer;  /*  Pointer to the edit pushbutton string. */

ppointer = _wpclsQueryEditString(somSelf);
```

wpclsQueryEditString Parameter - somSelf

somSelf (M_WPPalette *) - input
Pointer to the WPPalette class object.

wpclsQueryEditString Return Value - ppointer

ppointer (PSZ) - returns
Pointer to the edit pushbutton string.

wpclsQueryEditString - Parameters

somSelf (M_WPPalette *) - input
Pointer to the WPPalette class object.

ppointer (PSZ) - returns
Pointer to the edit pushbutton string.

wpclsQueryEditString - Usage

This method can be called at any time in order to determine the text of the edit pushbutton.

wpclsQueryEditString - How to Override

This method should be overridden in order to specify class-specific edit pushbutton text. The parent method is generally not called.

wpclsQueryEditString - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Glossary

wpclsQueryError

wpclsQueryError - Syntax

This class method queries the current error code held within a class object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;    /*  Pointer to the WPObject class object. */
ULONG           ulErrorId;  /*  ID of the last error. */

ulErrorId = _wpclsQueryError(somSelf);
```

wpclsQueryError Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsQueryError Return Value - ulErrorId

ulErrorId (ULONG) - returns
ID of the last error.

Last error that occurred when using this class object.

wpclsQueryError - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

ulErrorId (ULONG) - returns
ID of the last error.

Last error that occurred when using this class object.

wpclsQueryError - Remarks

When an error occurs within a class method and that method subsequently fails, the calling procedure can retrieve the error code for that failed method call by using the *wpclsQueryError* method. Note that the error code is always that of the *last method that failed*. A successful method does not modify the error code held within a class object. This method is analogous to the WinGetErrorInfo and the WinGetLastError function calls that are used by Presentation Manager applications to diagnose the reason for the previous failing call to a Presentation Manager function call.

The system-provided class methods will return error codes as defined in the header file PMERR.H. For example, the wpclsQueryObject method will normally log an error of WPERR_OBJECT_NOT_FOUND if it is unable to return an object pointer.

wpclsQueryError - Usage

This method should be called immediately after a class method has failed in order to diagnose why the failure occurred.

wpclsQueryError - How to Override

Never override this class method.

wpclsQueryError - Related Methods

Related Methods

- wpclsSetError

wpclsQueryError - Example Code

This example finds all occurrences of an object type in a folder.

```
 VOID  clsFindEverybody(SOMAny *self)
{
   BOOL      rc;                 /* Return from methods */
   HFIND     hFind       = 0;    /* Handle for repeated finds */
   CLASS     Class[2];           /* List of classes to look for */
   OBJECT    Object[10];         /* Objects returned from search */
   ULONG     ulCount     = 10;   /* Count of objects to find / or found */
   WPFolder  *folder     = 0;    /* Pointer to folder for search */
   ULONG     errorid     = 0;    /* Error returned from previous search */
   CHAR      sz[CCHMAXPATH];     /* Title string to look for */
   PSZ       title = sz;
   Class[0] = _WPFolder;
   Class[1] = NULL;

   /* Get the pointer to the desktop (since that is where we are looking) */

   folder = _wpclsQueryFolder( _WPFolder, "<WP_DESKTOP>", TRUE );

   _wpclsSetError( _WPObject, 0 );      /* Reset the error indicators */
   rc = 0;

         /* Attempt to find the first 10 objects matching our criteria */

   ulCount = 10;
   rc =_wpclsFindObjectFirst(_WPObject,     /* Class self pointer */
                             Class,         /* Pointer to array of class objects */
                             &hFind,        /* Handle for wpclsFindObjectNext */
                             NULL,          /* Find EVERYTHING */
                             folder,        /* Pointer to folder of interest */
                             TRUE,          /* Search all folders in tree */
                             NULL,          /* No extended criteria */
                             Object,        /* Array of object pointers */
                             &ulCount );    /* Number requested and number found */

   errorid = _wpclsQueryError( _WPObject);  /* Get errorid (may still be OK) */

   if ((!rc && (errorid == WPERR_BUFFER_OVERFLOW)) || rc) {

              /***** Process objects found here ******/

   } else {
      somPrintf(" _wpclsFindObjectFirst failed, errorid=%u\n", errorid );
      return;
   } /* endif */

   do {

      _wpclsSetError( _WPObject, 0 );    /* Reset error indicators */
      rc = 0;

             /* Look for next 10 objects */

      rc =_wpclsFindObjectNext(_WPObject,       /* Self */
                               hFind,           /* Handle from previous Find */
                               Object,          /* Objects found */
                               &ulCount );      /* Number to look for. On
                                                   return, number found */

      errorid = _wpclsQueryError( _WPObject );

      if ((!rc && (errorid == WPERR_BUFFER_OVERFLOW)) || rc) {

              /***** Process objects found here ******/
```

```
      } else {
            somPrintf("_wpclsFindObjectNext failed, errorid=%u\n", errorid);
            return;
      } /* endif */

   } while ( errorid == WPERR_BUFFER_OVERFLOW ); /* enddo */

   _wpclsSetError( _WPObject, 0 );    /* Reset error indicators */
   rc = 0;
               /* Indicate find complete */

   rc =_wpclsFindObjectEnd( _WPObject, hFind );

   if (!rc) {
     errorid = _wpclsQueryError( _WPObject );
     somPrintf( "_wpclsFindObjectEnd failed, errorid=%u\n", errorid );
     return;
   }
   return;
}
```

wpclsQueryError - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpclsQueryExtendedCriteria

wpclsQueryExtendedCriteria - Syntax

The wpclsQueryExtendedCriteria method is called to allow the specified class to specify the extended criteria to be used on a search for instances of this class.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;           /*  Pointer to the WPObject class object. */
PSZ             Title;             /*  Pointer to the title specification for objects to be searched. */
ULONG           ulSearchType;      /*  Flag indicating type of search to be performed. */
PVOID           ExtendedCriteria;  /*  Pointer to a buffer that contains the class-specific extended search crit
BOOL            fSuccess;          /*  Success indicator. */

fSuccess = wpclsQueryExtendedCriteria(somSelf,
             Title, ulSearchType, ExtendedCriteria);
```

wpclsQueryExtendedCriteria Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsQueryExtendedCriteria Parameter - Title

Title (PSZ) - input
Pointer to the title specification for objects to be searched.

This title may include the wildcard characters "*" and "?".

wpclsQueryExtendedCriteria Parameter - ulSearchType

ulSearchType (ULONG) - input
Flag indicating type of search to be performed.

SEARCH_ALL_FOLDERS
Search all folders.
SEARCH_THIS_FOLDER
Search current folder only.
SEARCH_THIS_TREE
Search the current folder tree only.

wpclsQueryExtendedCriteria Parameter - ExtendedCriteria

ExtendedCriteria (PVOID) - output
Pointer to a buffer that contains the class-specific extended search criteria.

This buffer can be passed to the wpDoesObjectMatch method.

wpclsQueryExtendedCriteria Return Value - fSuccess

fSuccess (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
An error occurred or the user canceled the search operation.

wpclsQueryExtendedCriteria - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

Title (PSZ) - input
Pointer to the title specification for objects to be searched.

This title may include the wildcard characters "*" and "?".

ulSearchType (ULONG) - input
Flag indicating type of search to be performed.

SEARCH_ALL_FOLDERS
Search all folders.
SEARCH_THIS_FOLDER
Search current folder only.
SEARCH_THIS_TREE
Search the current folder tree only.

ExtendedCriteria (PVOID) - output
Pointer to a buffer that contains the class-specific extended search criteria.

This buffer can be passed to the wpDoesObjectMatch method.

fSuccess (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
An error occurred or the user canceled the search operation.

wpclsQueryExtendedCriteria - Remarks

This method is called by the system during the processing of the Find and folder Include facilities.

This method typically prompts the user to select the extended criteria.

wpclsQueryExtendedCriteria - Usage

This method can be called at any time in order to determine the extended search criteria to be used for finding objects of this class.

wpclsQueryExtendedCriteria - How to Override

This method should be overridden by classes which introduce extended search criteria for use by the Find and folder Include facilities.

wpclsQueryExtendedCriteria - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Glossary

wpclsQueryFolder

wpclsQueryFolder - Syntax

This method is called to get a pointer to a folder object that corresponds to a given file system location.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;    /*  Pointer to the WPObject class object. */
PSZ             pLocation;  /*  Folder location. */
BOOL            fLock;      /*  Lock object flag. */
WPFolder       *Success;    /*  Success indicator. */

Success = _wpclsQueryFolder(somSelf, pLocation,
            fLock);
```

wpclsQueryFolder Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsQueryFolder Parameter - pLocation

pLocation (PSZ) - input
Folder location.

This value can be in any of the following formats:

- Predefined object IDs of system folders:

"<WP_NOWHERE>"	The hidden folder.
"<LOCATION_DESKTOP>"	The currently active desktop.
"<WP_OS2SYS>"	The System folder.
"<WP_TEMPS>"	The Templates folder.
"<WP_CONFIG>"	The System Setup folder.
"<WP_START>"	The Startup folder.
"<WP_INFO>"	The Information folder.
"<WP_DRIVES>"	The Drives folder.

- Real name specified as a fully qualified path name.

wpclsQueryFolder Parameter - fLock

fLock (BOOL) - input
Lock object flag.

If this flag is false, the newly created object will be made dormant whenever the object and the folder containing the object are closed.
If this flag is true, the new flag will remain active until the caller issues the wpUnlockObject method on it.

wpclsQueryFolder Return Value - Success

Success (WPFolder *) - returns
Success indicator.

NULL	Error occurred
Other	Pointer to a folder object.

wpclsQueryFolder - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

pLocation (PSZ) - input
Folder location.

This value can be in any of the following formats:

- Predefined object IDs of system folders:

"<WP_NOWHERE>"	The hidden folder.
"<LOCATION_DESKTOP>"	The currently active desktop.
"<WP_OS2SYS>"	The System folder.
"<WP_TEMPS>"	The Templates folder.
"<WP_CONFIG>"	The System Setup folder.
"<WP_START>"	The Startup folder.
"<WP_INFO>"	The Information folder.
"<WP_DRIVES>"	The Drives folder.

- Real name specified as a fully qualified path name.

fLock (BOOL) - input
Lock object flag.

If this flag is false, the newly created object will be made dormant whenever the object and the folder containing the object are closed.
If this flag is true, the new flag will remain active until the caller issues the wpUnlockObject method on it.

Success (WPFolder *) - returns
Success indicator.

NULL	Error occurred
Other	Pointer to a folder object.

wpclsQueryFolder - Remarks

To obtain a real name from an object pointer, the wpQueryRealName method should be called.

wpclsQueryFolder - Usage

This method can be called at any time in order to determine the object pointer for a folder.

wpclsQueryFolder - How to Override

This method is generally not overridden.

wpclsQueryFolder - Example Code

```
#define INCL_WINWORKPLACE
#include <os2.h>

SOMAny    *objDesktop;
SOMAny    *objSysShadow;


     /* Get the object pointer of the desktop folder */

objDesktop = _wpclsQueryFolder( _WPFolder
                              , "<WP_DESKTOP>"     /* Folder location */
                              ,TRUE);              /* Keep active */

     /* Create a shadow of the desktop */

objSysShadow = _wpCreateShadowObject( self
                                    , objDesktop   /* Object to shadow */
                                    ,FALSE);       /* Make dormant when folder
                                                      closes */
```

wpclsQueryFolder - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Glossary

wpclsQueryIcon

wpclsQueryIcon - Syntax

This method is called to allow the class object to specify the default icon to be used for its instances.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;      /*  Pointer to the WPObject class object. */
HPOINTER        hptrSuccess;  /*  Success indicator. */

hptrSuccess = _wpclsQueryIcon(somSelf);
```

wpclsQueryIcon Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsQueryIcon Return Value - hptrSuccess

hptrSuccess (HPOINTER) - returns
Success indicator.

NULL	Error occurred
Other	Handle to an icon.

wpclsQueryIcon - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

hptrSuccess (HPOINTER) - returns
Success indicator.

NULL	Error occurred
Other	Handle to an icon.

wpclsQueryIcon - Remarks

The class default icon can be loaded on wpclsInitData and freed on wpclsUnInitData.

wpclsQueryIcon - Usage

This method can be called at any time in order to determine the default icon for instances of this class.

wpclsQueryIcon - How to Override

This method is overridden in order to change the default icon for an instance of the class.

wpclsQueryIcon - Related Methods

Related Methods

- wpclsQueryIcon
- wpQueryIconData
- wpSetIcon
- wpSetIconData
- wpQueryIcon
- wpclsQueryIconData

wpclsQueryIcon - Example Code

This example returns the handle to an icon.

```
SOM_Scope HPOINTER   SOMLINK pviewM_wpclsQueryIcon(M_PictureViewer *somSelf)
{
    /* M_PictureViewerData *somThis = M_PictureViewerGetData(somSelf); */
    M_PictureViewerMethodDebug("M_PictureViewer","pviewM_wpclsQueryIcon");

/*    return (parent_wpclsQueryIcon(somSelf));   don't call parent method */
      return ( _hIcon );                         /* Return own Icon */
}
```

wpclsQueryIcon - Topics

Select an item:
Syntax
Parameters
Returns
Remarks

How to Override
Usage
Example Code
Related Methods
Glossary

wpclsQueryIconData

wpclsQueryIconData - Syntax

This class method allows the system to build the class default icon for a given class.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;    /*  Pointer to the class object for which you want the class icon data. */
PICONINFO       pIconInfo;  /*  Pointer to a buffer. */
ULONG           ulReturn;   /*  Buffer size or number of bytes required to hold the output data. */

ulReturn = _wpclsQueryIconData(somSelf, pIconInfo);
```

wpclsQueryIconData Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the class object for which you want the class icon data.

wpclsQueryIconData Parameter - pIconInfo

pIconInfo (PICONINFO) - output
Pointer to a buffer.

Pointer to the buffer large enough to contain an ICONINFO data structure followed by the data needed to describe the icon.

If *pIconInfo* is NULL, the return value is the required size of the buffer.

wpclsQueryIconData Return Value - ulReturn

ulReturn (ULONG) - returns
Buffer size or number of bytes required to hold the output data.

Depending of the contents of the *plconInfo* parameter, *ulReturn* contains one of the following:

plconInfo	*ulReturn*
NULL	Number of bytes required to hold the output data for this class.
Not NULL	Number of bytes written into the buffer.

If *ulReturn* is 0, an error occurred.

wpclsQueryIconData - Parameters

somSelf (M_WPObject *) - input
Pointer to the class object for which you want the class icon data.

plconInfo (PICONINFO) - output
Pointer to a buffer.

Pointer to the buffer large enough to contain an ICONINFO data structure followed by the data needed to describe the icon.

If *plconInfo* is NULL, the return value is the required size of the buffer.

ulReturn (ULONG) - returns
Buffer size or number of bytes required to hold the output data.

Depending of the contents of the *plconInfo* parameter, *ulReturn* contains one of the following:

plconInfo	*ulReturn*
NULL	Number of bytes required to hold the output data for this class.
Not NULL	Number of bytes written into the buffer.

If *ulReturn* is 0, an error occurred.

wpclsQueryIconData - Remarks

If NULL is passed for the *plconInfo* parameter, the caller is asking for the size of the ICONINFO buffer needed for this class (usually for memory allocation purposes). Otherwise, the *plconInfo* parameter can always be assumed to be large enough to accommodate the ICONINFO and the variable data for this class.

Note that the ICONINFO structure allows you to specify the default icon in three different ways:

- Block of binary data
- Icon file name
- Module name and resource identifier

However, only one mechanism need be supported by any given class. For example, a caller cannot request one of the three formats by prefilling the ICONINFO structure.

wpclsQueryIconData - Usage

This method may be called at any time. Typically, it would not be useful for another object class to make calls to this method.

wpclsQueryIconData - How to Override

Workplace classes that wish to have a unique class default icon must override this method and fill out the appropriate fields within the ICONINFO structure. In addition, the correct size for the ICONINFO must always be returned.

wpclsQueryIconData - Related Methods

Related Methods

- wpQueryIcon
- wpQueryIconData
- wpSetIcon
- wpSetIconData
- wpclsQueryIcon

wpclsQueryIconData - Example Code

This example returns information about the icon associated with the class if *pIconInfo* is not NULL.

```
SOM_Scope ULONG SOMLINK CalendarM_wpclsQueryIconData(M_Calendar *somSelf,
                                                     PICONINFO pIconInfo)
{
  M_CalendarData *somThis = M_CalendarGetData(somSelf);
  M_CalendarMethodDebug("M_Calendar","myfM_wpclsQueryIconData");

  if (pIconInfo)
  {
    pIconInfo->fFormat = ICON_RESOURCE;
    pIconInfo->hmod    = _clsQueryModuleHandle(somSelf);
    pIconInfo->resid   = ID_ICON;
  } /* endif */
  return (sizeof(ICONINFO));
}
```

wpclsQueryIconData - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpclsQueryIconDataN

wpclsQueryIconDataN - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This class method allows the system to build the class default animation icon for a given class.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPFolder     *somSelf;       /*  Pointer to the WPFolder class object. */
PICONINFO       pIconInfo;     /*  Pointer to the icon information. */
ULONG           ulcbIconInfo;  /*  Size of the buffer needed to store the ICONINFO data returned by this class ob
ULONG           ulIconIndex;   /*  Animation icon index. */

_wpclsQueryIconDataN(somSelf, pIconInfo, ulcbIconInfo,
                ulIconIndex);
```

wpclsQueryIconDataN Parameter - somSelf

somSelf (M_WPFolder *) - input
Pointer to the WPFolder class object.

wpclsQueryIconDataN Parameter - pIconInfo

pIconInfo (PICONINFO) - input
Pointer to the icon information.

wpclsQueryIconDataN Parameter - ulcbIconInfo

ulcbIconInfo (ULONG) - input
Size of the buffer needed to store the ICONINFO data returned by this class object.

wpclsQueryIconDataN Parameter - ulIconIndex

ulIconIndex (ULONG) - input
Animation icon index.

This value must be 1.

wpclsQueryIconDataN - Parameters

somSelf (M_WPFolder *) - input
Pointer to the WPFolder class object.

pIconInfo (PICONINFO) - input
Pointer to the icon information.

ulcbIconInfo (ULONG) - input
Size of the buffer needed to store the ICONINFO data returned by this class object.

ulIconIndex (ULONG) - input
Animation icon index.

This value must be 1.

wpclsQueryIconDataN - Remarks

If NULLHANDLE is passed for the *pIconInfo* parameter, the caller is asking for the size of the ICONINFO buffer needed for this class usually for memory allocation purposes. Otherwise, the *pIconInfo* parameter can always be assumed to be large enough to accommodate the ICONINFO for this class.

Note that the ICONINFO structure allows you to specify the default icon in three different ways:

- An icon filename
- A module name and resource identifier
- A block of binary data

However, only one mechanism needs to be supported for any given class. For example, a caller cannot request one of the three formats by prefilling the ICONINFO structure.

wpclsQueryIconDataN - Usage

This method may be called at any time. Typically, it would not be useful for another object class to make calls to this method.

wpclsQueryIconDataN - How to Override

Workplace classes that wish to have a unique class default animation icon must override this method and fill out the appropriate fields within the ICONINFO structure. In addition, the correct size for the ICONINFO must always be returned.

wpclsQueryIconDataN - Related Methods

Related Methods

- wpQueryIcon
- wpQueryIconData
- wpSetIcon
- wpSetIconData
- wpclsQueryIconN

wpclsQueryIconDataN - Topics

Select an item:
Syntax
Parameters
Remarks
How to Override
Usage
Related Methods
Glossary

wpclsQueryIconN

wpclsQueryIconN - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This method is called to allow the class object to specify the default animation icon to be used for its instances.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPFolder     *somSelf;      /*  Pointer to the WPFolder class object. */
ULONG           ulIconIndex;  /*  Animation icon index. */
HPOINTER        hptr;         /*  Handle to the animation icon. */

hptr = _wpclsQueryIconN(somSelf, ulIconIndex);
```

wpclsQueryIconN Parameter - somSelf

somSelf (M_WPFolder *) - input
Pointer to the WPFolder class object.

wpclsQueryIconN Parameter - ulIconIndex

ulIconIndex (ULONG) - input
Animation icon index.

This value must be 1.

wpclsQueryIconN Return Value - hptr

hptr (HPOINTER) - returns
Handle to the animation icon.

A return value of NULL indicates that there is no animation icon.

wpclsQueryIconN - Parameters

somSelf (M_WPFolder *) - input
Pointer to the WPFolder class object.

ulIconIndex (ULONG) - input
Animation icon index.

This value must be 1.

hptr (HPOINTER) - returns
Handle to the animation icon.

A return value of NULL indicates that there is no animation icon.

wpclsQueryIconN - Remarks

This method is very similar to wpclsQueryIcon. The difference is that wpclsQueryIcon returns the handle for a normal object icon (the closed folder icon), while wpclsQueryIconN returns the handle for the animation icon (the open folder icon).

wpclsQueryIconN - Usage

This method can be called at any time.

wpclsQueryIconN - How to Override

This method generally not overridden.

wpclsQueryIconN - Related Methods

Related Methods

- wpclsQueryIcon
- wpQueryIconData

- wpSetIcon
- wpSetIconData
- wpQueryIcon
- wpclsQueryIconData
- wpclsQueryIconDataN

wpclsQueryIconN - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpclsQueryInstanceClass

wpclsQueryInstanceClass - Syntax

This class method returns the name of a class that this object can become.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPFileSystem     *somSelf;   /*  Pointer to the WPFileSystem class object. */
PSZ                 pszClass;  /*  String containing the name of the new class for files of this class. */

pszClass = wpclsQueryInstanceClass(somSelf);
```

wpclsQueryInstanceClass Parameter - somSelf

somSelf (M_WPFileSystem *) - input
Pointer to the WPFileSystem class object.

wpclsQueryInstanceClass Return Value - pszClass

pszClass (PSZ) - returns
String containing the name of the new class for files of this class.

wpclsQueryInstanceClass - Parameters

somSelf (M_WPFileSystem *) - input
Pointer to the WPFileSystem class object.

pszClass (PSZ) - returns
String containing the name of the new class for files of this class.

wpclsQueryInstanceClass - Usage

This method can be called at any time to find out what class an object of this class can become.

wpclsQueryInstanceClass - How to Override

This method should be overridden in any object class that allows instances of the object class to change to another object class by using the Become page of the Properties notebook of the object.

wpclsQueryInstanceClass - Related Methods

Related Methods

- wpclsQueryInstanceFilter
- wpclsQueryInstanceType

wpclsQueryInstanceClass - Example Code

This example shows how "MyClass" allows instances of "MyClass" to become plain WPDataFile objects.

```
SOM_Scope PSZ SOMLINK mcM_wpclsQueryInstanceClass(M_MyClass *self)
{
   /* Allow an instance of MyClass to become a WPDataFile object  */
   return "WPDataFile";
}
```

wpclsQueryInstanceClass - Topics

Select an item:
Syntax

Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpclsQueryInstanceFilter

wpclsQueryInstanceFilter - Syntax

This method is called to allow the class object to specify the file title filters for instances of its class.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPFileSystem     *somSelf;   /*  Pointer to the WPFileSystem class object. */
PSZ                 psuccess;  /*  Success indicator. */

psuccess = _wpclsQueryInstanceFilter(somSelf);
```

wpclsQueryInstanceFilter Parameter - somSelf

somSelf (M_WPFileSystem *) - input
Pointer to the WPFileSystem class object.

wpclsQueryInstanceFilter Return Value - psuccess

psuccess (PSZ) - returns
Success indicator.

NULL	Error occurred.
Other	A pointer to a string containing file title filters. This string can contain several file title filters separated by commas. For example: `"*.TXT, *.DOC"`

wpclsQueryInstanceFilter - Parameters

somSelf (M_WPFileSystem *) - input
Pointer to the WPFileSystem class object.

psuccess (PSZ) - returns
Success indicator.

NULL	Error occurred.
Other	A pointer to a string containing file title filters. This string can contain several file title filters separated by commas. For example: `"*.TXT, *.DOC"`

wpclsQueryInstanceFilter - Remarks

It is important that the values returned by this class method are restricted to class-specific filters. For example, returning a filter of *"*.*"* could effectively make the system unstable.

wpclsQueryInstanceFilter - Usage

This method can be called at any time in order to determine which file title filters are used to determine instances of this class.

wpclsQueryInstanceFilter - How to Override

This method should be overridden in order to automatically designate file objects as instances of this class. The value returned by the override method will replace the current title filter string which is used to designate instances. If the parent method is called, it should be called first.

wpclsQueryInstanceFilter - Related Methods

Related Methods

- wpclsQueryInstanceType

wpclsQueryInstanceFilter - Example Code

This example shows how to designate files ending in an extension of .GIF as being associated with an instance of a GIFPicture object.

```
SOM_Scope PSZ   SOMLINK gifpicM_wpclsQueryInstanceFilter(M_GIFPicture *somSelf)
{
    /* M_GIFPictureData *somThis = M_GIFPictureGetData(somSelf); */
    M_GIFPictureMethodDebug("M_GIFPicture","gifpicM_wpclsQueryInstanceFilter");

/*    return (parent_wpclsQueryInstanceFilter(somSelf));   Don't call parent */

      return ( "*.GIF" );
```

}

wpclsQueryInstanceFilter - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpclsQueryInstanceType

wpclsQueryInstanceType - Syntax

This method is called to allow the class object to specify the file types for instances of its class.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPFileSystem     *somSelf;   /*  Pointer to the WPFileSystem class object. */
PSZ                 psuccess;  /*  Success indicator. */

psuccess = _wpclsQueryInstanceType(somSelf);
```

wpclsQueryInstanceType Parameter - somSelf

somSelf (M_WPFileSystem *) - input
Pointer to the WPFileSystem class object.

wpclsQueryInstanceType Return Value - psuccess

psuccess (PSZ) - returns
Success indicator.

NULL Error occurred.

Other	A pointer to a string containing file types. This string can contain several file types separated by commas. For example: `"Plain Text,C Code"`

wpclsQueryInstanceType - Parameters

somSelf (M_WPFileSystem *) - input
Pointer to the WPFileSystem class object.

psuccess (PSZ) - returns
Success indicator.

NULL	Error occurred.
Other	A pointer to a string containing file types. This string can contain several file types separated by commas. For example: `"Plain Text,C Code"`

wpclsQueryInstanceType - Remarks

It is recommended that object classes define their own special type strings.

wpclsQueryInstanceType - Usage

This method can be called at any time in order to determine which file types are used to determine instances of this class.

wpclsQueryInstanceType - How to Override

This method should be overridden in order to automatically designate file objects as instances of this class. The value returned by the override method will replace the current type string which is used to designate instances. If the parent method is called, it should be called first.

wpclsQueryInstanceType - Related Methods

Related Methods

- wpclsQueryInstanceFilter

wpclsQueryInstanceType - Example Code

This example associates two file types as being associated with the Picture object.

```
SOM_Scope PSZ   SOMLINK PictureM_wpclsQueryInstanceType(M_Picture *somSelf)
{
    /* M_PictureData *somThis = M_PictureGetData(somSelf); */
    M_PictureMethodDebug("M_Picture","PictureM_wpclsQueryInstanceType");

/*  return (parent_wpclsQueryInstanceType(somSelf));  no call to parent */

    return ( "JPG File,GIF File" );
}
```

wpclsQueryInstanceType - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpclsQueryObject

wpclsQueryObject - Syntax

This class method returns the object pointer for a given persistent object handle.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;  /*  Pointer to the WPObject class object. */
HOBJECT         hObject;  /*  Handle for a given object instance. */
WPObject       *object;   /*  Pointer to the object. */

object = _wpclsQueryObject(somSelf, hObject);
```

wpclsQueryObject Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsQueryObject Parameter - hObject

hObject (HOBJECT) - input
Handle for a given object instance.

wpclsQueryObject Return Value - object

object (WPObject *) - returns
Pointer to the object.

The pointer to the object that corresponds to the given object handle or NULLHANDLE if that object no longer exists.

wpclsQueryObject - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

hObject (HOBJECT) - input
Handle for a given object instance.

object (WPObject *) - returns
Pointer to the object.

The pointer to the object that corresponds to the given object handle or NULLHANDLE if that object no longer exists.

wpclsQueryObject - Remarks

All workplace objects can be uniquely identified by a persistent object handle within a given machine. Once an object handle is obtained for an object (using the wpQueryHandle instance method), the handle can be used at any subsequent time even if the system was powered off in the meantime, provided that the object instance has not been destroyed. This method is intended for use by objects that wish to communicate with other objects using method calls. At any point in time, an object can reestablish contact with another object by using this method. For example, a WPShadow object will use this method to get the object pointer to the object it is currently shadowing when it is made awake. Note that the returned object is locked so that the called object can access the returned pointer without the risk of the object being made dormant. When the object pointer is no longer needed, the wpUnlockObject method should be invoked to permit the system to make that object dormant when all other locks on it have been released.

Using HOBJECT for .INI files or files in which an application uses a rename/save/delete sequence is not supported.

wpclsQueryObject - Usage

This method can be called at any time.

wpclsQueryObject - How to Override

This class method is generally not overridden.

wpclsQueryObject - Related Methods

Related Methods

- wpclsObjectFromHandle
- wpclsQueryObjectFromFrame
- wpclsQueryObjectFromPath

wpclsQueryObject - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpclsQueryObjectFromFrame

wpclsQueryObjectFromFrame - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This class method returns a pointer to the object associated with the specified frame window handle.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPDesktop     *somSelf;    /*  Pointer to the WPDesktop class object. */
HWND             hwndFrame;  /*  Handle to the top-level frame window. */
WPObject        *Object;     /*  Pointer to the object associated with hwndFrame. */

Object = _wpclsQueryObjectFromFrame(somSelf,
           hwndFrame);
```

wpclsQueryObjectFromFrame Parameter - somSelf

somSelf (M_WPDesktop *) - input
Pointer to the WPDesktop class object.

wpclsQueryObjectFromFrame Parameter - hwndFrame

hwndFrame (HWND) - input
Handle to the top-level frame window.

wpclsQueryObjectFromFrame Return Value - Object

Object (WPObject *) - returns
Pointer to the object associated with *hwndFrame* .

This parameter contains a pointer to the object of which its view has the given frame window handle as its top-level frame window handle.

A return value of NULL indicates that either an associated object could not be found or an error occurred.

wpclsQueryObjectFromFrame - Parameters

somSelf (M_WPDesktop *) - input
Pointer to the WPDesktop class object.

hwndFrame (HWND) - input
Handle to the top-level frame window.

Object (WPObject *) - returns
Pointer to the object associated with *hwndFrame* .

This parameter contains a pointer to the object of which its view has the given frame window handle as its top-level frame window handle.

A return value of NULL indicates that either an associated object could not be found or an error occurred.

wpclsQueryObjectFromFrame - Remarks

This method returns the object for which the handle is a view rather than the application implementing the view.

For example, if the *hwndFrame* of a folder's view is passed in, this method returns the folder object. If the *hwndFrame* of the system editor is passed in, this method returns the object which is currently being viewed by the editor. In this case, if an associated object cannot be found, the system editor program object is returned. For instance, if the system editor is launched from the command line and its *hwndFrame* is passed in, the editor program object is returned.

wpclsQueryObjectFromFrame - Usage

This method can be called at any time.

wpclsQueryObjectFromFrame - How to Override

This method should not be overridden.

wpclsQueryObjectFromFrame - Related Methods

Related Methods

- wpclsQueryObject
- wpclsObjectFromHandle
- wpclsQueryObjectFromPath

wpclsQueryObjectFromFrame - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpclsQueryObjectFromPath

wpclsQueryObjectFromPath - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This class method returns the object pointer for a file or directory.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPFileSystem     *somSelf;     /*  Pointer to the WPFileSystem class object. */
PSZ                 pszFQPath;   /*  String containing the fully qualified path name (or object ID). */
WPObject           *pObject;     /*  Pointer to the object that represents the file or directory specified in psz
```

```
pObject = _wpclsQueryObjectFromPath(somSelf,
            pszFQPath);
```

wpclsQueryObjectFromPath Parameter - somSelf

somSelf (M_WPFileSystem *) - input
Pointer to the WPFileSystem class object.

wpclsQueryObjectFromPath Parameter - pszFQPath

pszFQPath (PSZ) - input
String containing the fully qualified path name (or object ID).

wpclsQueryObjectFromPath Return Value - pObject

pObject (WPObject *) - returns
Pointer to the object that represents the file or directory specified in *pszFQPath* .

wpclsQueryObjectFromPath - Parameters

somSelf (M_WPFileSystem *) - input
Pointer to the WPFileSystem class object.

pszFQPath (PSZ) - input
String containing the fully qualified path name (or object ID).

pObject (WPObject *) - returns
Pointer to the object that represents the file or directory specified in *pszFQPath* .

wpclsQueryObjectFromPath - Remarks

The object is initially locked, so there is no risk of the object being made dormant before the object pointer is obtained.

wpclsQueryObjectFromPath - How to Override

This method is generally not overridden.

wpclsQueryObjectFromPath - Related Methods

Related Methods

- wpclsQueryObject

wpclsQueryObjectFromPath - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpclsQueryOpenFolders

wpclsQueryOpenFolders - Syntax

This class method enumerates all open folders for the specified class.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPFolder     *somSelf;     /*  Pointer to the WPFolder class object. */
WPFolder       *Folder;      /*  Pointer to a folder object. */
ULONG           ulOption;    /*  Flag indicating the folder to query. */
BOOL            fLock;       /*  Lock the object flag. */
WPFolder       *FldrObject;  /*  Pointer to the specified folder object. */

FldrObject = _wpclsQueryOpenFolders(somSelf,
               Folder, ulOption, fLock);
```

wpclsQueryOpenFolders Parameter - somSelf

somSelf (M_WPFolder *) - input
Pointer to the WPFolder class object.

wpclsQueryOpenFolders Parameter - Folder

Folder (WPFolder *) - input
Pointer to a folder object.

This field is ignored unless QC_NEXT is specified in *ulOption* .

wpclsQueryOpenFolders Parameter - ulOption

ulOption (ULONG) - input
Flag indicating the folder to query.

Possible values are described in the following list:

QC_FIRST
Return the first open folder in the open-folder list. This value is specified on the class object if the passed object is not currently open.

QC_LAST
Return the last open folder.

QC_NEXT
Return the next open folder after *Folder* in the list.

wpclsQueryOpenFolders Parameter - fLock

fLock (BOOL) - input
Lock the object flag.

Possible values are described in the following list:

TRUE
The new flag will remain active until the caller issues wpUnlockObject on it.

FALSE
The newly created object will be made dormant whenever the object and the folder containing the object are closed.

wpclsQueryOpenFolders Return Value - FldrObject

FldrObject (WPFolder *) - returns
Pointer to the specified folder object.

A return value of NULL indicates that either an error occurred or QC_NEXT was requested on the last folder.

wpclsQueryOpenFolders - Parameters

somSelf (M_WPFolder *) - input
Pointer to the WPFolder class object.

Folder (WPFolder *) - input
Pointer to a folder object.

This field is ignored unless QC_NEXT is specified in *ulOption*.

ulOption (ULONG) - input
Flag indicating the folder to query.

Possible values are described in the following list:

QC_FIRST
Return the first open folder in the open-folder list. This value is specified on the class object if the passed object is not currently open.

QC_LAST
Return the last open folder.

QC_NEXT
Return the next open folder after *Folder* in the list.

fLock (BOOL) - input
Lock the object flag.

Possible values are described in the following list:

TRUE
The new flag will remain active until the caller issues wpUnlockObject on it.

FALSE
The newly created object will be made dormant whenever the object and the folder containing the object are closed.

FldrObject (WPFolder *) - returns
Pointer to the specified folder object.

A return value of NULL indicates that either an error occurred or QC_NEXT was requested on the last folder.

wpclsQueryOpenFolders - Usage

This method can be called at any time.

wpclsQueryOpenFolders - How to Override

This method is generally not overridden.

wpclsQueryOpenFolders - Related Methods

Related Methods

- wpContainsFolders

wpclsQueryOpenFolders - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpclsQuerySetting

wpclsQuerySetting - Syntax

This method returns the class object settings from the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPAbstract     *somSelf;         /*  Pointer to the WPAbstract class object. */
PSZ               pszSetting;      /*  Name of the setting that is to be queried. */
PVOID             pValue;          /*  Buffer in which settings values are returned. */
ULONG             ulValueLen;      /*  Length of pValue. */
ULONG             ulReturnedData;  /*  Length of data returned in pValue. */

ulReturnedData = _wpclsQuerySetting(somSelf,
                   pszSetting, pValue, ulValueLen);
```

wpclsQuerySetting Parameter - somSelf

somSelf (M_WPAbstract *) - input
Pointer to the WPAbstract class object.

wpclsQuerySetting Parameter - pszSetting

pszSetting (PSZ) - input
Name of the setting that is to be queried.

wpclsQuerySetting Parameter - pValue

pValue (PVOID) - in/out
Buffer in which settings values are returned.

If this parameter is NULL, only the length of the value of the setting is returned.

wpclsQuerySetting Parameter - ulValueLen

ulValueLen (ULONG) - input
Length of *pValue* .

wpclsQuerySetting Return Value - ulReturnedData

ulReturnedData (ULONG) - returns
Length of data returned in *pValue* .

A return value of 0 indicates an error occurred, in which case no data is returned in *pValue* .

wpclsQuerySetting - Parameters

somSelf (M_WPAbstract *) - input
Pointer to the WPAbstract class object.

pszSetting (PSZ) - input
Name of the setting that is to be queried.

pValue (PVOID) - in/out
Buffer in which settings values are returned.

If this parameter is NULL, only the length of the value of the setting is returned.

ulValueLen (ULONG) - input
Length of *pValue* .

ulReturnedData (ULONG) - returns
Length of data returned in *pValue* .

A return value of 0 indicates an error occurred, in which case no data is returned in *pValue* .

wpclsQuerySetting - Remarks

This is a virtual method and must be overridden by a subclass. Each subclass documents the possible values for *pszSetting* and *pValue* . These settings and values are specific for each subclass. Note that all settings have safe defaults, so it is never necessary to pass unnecessary settings to an object.

For a list of settings and their associated values, see the individual classes.

wpclsQuerySetting - Usage

This method should never be called. It is a virtual method and must be overridden by a subclass.

wpclsQuerySetting - How to Override

This is a virtual method and must be overridden by a subclass.

wpclsQuerySetting - Related Methods

Related Methods

- wpclsSetSetting

wpclsQuerySetting - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpclsQuerySettingsPageSize

wpclsQuerySettingsPageSize - Syntax

This class method returns the default size of a Settings page in dialog units for instances of this object class.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;  /*  Pointer to the WPObject class object. */
PSIZEL          pSizl;    /*  Pointer to the default Settings page dimensions. */
BOOL            rc;       /*  Success indicator. */

rc = _wpclsQuerySettingsPageSize(somSelf,
       pSizl);
```

wpclsQuerySettingsPageSize Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsQuerySettingsPageSize Parameter - pSizl

pSizl (PSIZEL) - in/out
Pointer to the default Settings page dimensions.

A pointer to the SIZEL structure containing the *cx* and *cy* dimensions of the default Settings page for this class.

wpclsQuerySettingsPageSize Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	The method class call was successful.
FALSE	The method class call was unsuccessful.

wpclsQuerySettingsPageSize - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

pSizl (PSIZEL) - in/out
Pointer to the default Settings page dimensions.

A pointer to the SIZEL structure containing the *cx* and *cy* dimensions of the default Settings page for this class.

rc (BOOL) - returns
Success indicator.

TRUE	The method class call was successful.
FALSE	The method class call was unsuccessful.

wpclsQuerySettingsPageSize - Remarks

An object class that has abnormally shaped Settings pages (for example, the pages may be very wide) can use this method to ensure that, when a Settings notebook is initially displayed for an instance of the object class, that Settings notebook will be of a suitable size to view the whole Settings page without clipping it. The size specified by this method is used only the very first time that a Settings notebook is displayed; thereafter, the Settings notebook size and position will be saved by the system on a per-object basis.

wpclsQuerySettingsPageSize - Usage

This method can be called at any time.

wpclsQuerySettingsPageSize - How to Override

Object classes with unusual size requirements for their Settings pages must override this class method and change the *cx* and *cy* values as necessary to accommodate their Settings page sizes.

wpclsQuerySettingsPageSize - Related Methods

Related Methods

- wpOpen

wpclsQuerySettingsPageSize - Example Code

This example changes the size of the Settings page for the Person object. This size is only used when the settings are initially displayed. After that, the size and position are saved on a per-object basis.

```
SOM_Scope BOOL   SOMLINK PersonM_wpclsQuerySettingsPageSize(M_Person *somSelf,
                PSIZEL pSizl)

{
    /* M_PersonData *somThis = M_PersonGetData(somSelf); */
    M_PersonMethodDebug("M_Person","PersonM_wpclsQuerySettingsPageSize");

    pSizl->Cx = 600;
    pSizl->Cy = 1200;

/*    return (parent_wpclsQuerySettingsPageSize(somSelf,pSizl));  */

    return ( TRUE );
}
```

wpclsQuerySettingsPageSize - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods

wpclsQueryStyle

wpclsQueryStyle - Syntax

This method is called to allow the class object to specify the default object class style for its instances.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;   /*  Pointer to the WPObject class object. */
ULONG           ulReturn;  /*  Class style for this object. */

ulReturn = _wpclsQueryStyle(somSelf);
```

wpclsQueryStyle Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsQueryStyle Return Value - ulReturn

ulReturn (ULONG) - returns
Class style for this object.

CLSSTYLE_DONTTEMPLATE
Do not allow a create-template operation on objects of this class.

CLSSTYLE_NEVERCOPY
Do not allow a copy operation on objects of this class.

CLSSTYLE_NEVERDELETE
Do not allow a delete operation on objects of this class.

CLSSTYLE_NEVERDRAG
Do not allow a drag operation on objects of this class.

CLSSTYLE_NEVERDROPON
Do not allow a dropon operation on objects of this class.

CLSSTYLE_NEVERLINK
Do not allow a create-shadow operation on objects of this class.

CLSSTYLE_NEVERMOVE
Do not allow a move operation on objects of this class.

CLSSTYLE_NEVERPRINT
Do not allow a print of this object.

CLSSTYLE_NEVERRENAME
Do not allow the renaming of objects of this class.

CLSSTYLE_NEVERSETTINGS
Do not allow a settings operation on objects of this class.

CLSSTYLE_NEVERVISIBLE
Make instances of this class always invisible.

wpclsQueryStyle - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

ulReturn (ULONG) - returns
Class style for this object.

CLSSTYLE_DONTTEMPLATE
Do not allow a create-template operation on objects of this class.

CLSSTYLE_NEVERCOPY
Do not allow a copy operation on objects of this class.

CLSSTYLE_NEVERDELETE
Do not allow a delete operation on objects of this class.

CLSSTYLE_NEVERDRAG
Do not allow a drag operation on objects of this class.

CLSSTYLE_NEVERDROPON
Do not allow a dropon operation on objects of this class.

CLSSTYLE_NEVERLINK
Do not allow a create-shadow operation on objects of this class.

CLSSTYLE_NEVERMOVE
Do not allow a move operation on objects of this class.

CLSSTYLE_NEVERPRINT
Do not allow a print of this object.

CLSSTYLE_NEVERRENAME
Do not allow the renaming of objects of this class.

CLSSTYLE_NEVERSETTINGS
Do not allow a settings operation on objects of this class.

CLSSTYLE_NEVERVISIBLE
Make instances of this class always invisible.

wpclsQueryStyle - Remarks

When an instance is initially created, it has the same object style (OBJSTYLE_xxx) flags as its class style (CLSSTYLE_xxx).

wpclsQueryStyle - Usage

This method can be called at any time in order to determine the default style for instances of this class.

wpclsQueryStyle - How to Override

This method should be overridden in order to modify the default object style for instances of this class.

wpclsQueryStyle - Related Methods

Related Methods

- wpclsQueryStyle
- wpQueryStyle
- wpSetStyle

wpclsQueryStyle - Example Code

This sample shows how to remove the Print option from the menu for BigObject.

```
SOM_Scope ULONG   SOMLINK BigObjectM_wpclsQueryStyle(M_BigObject *somSelf)
{
    /* M_BigObjectData *somThis = M_BigObjectGetData(somSelf); */
    M_BigObjectMethodDebug("M_BigObject","BigObjectM_wpclsQueryStyle");

    return ( parent_wpclsQueryStyle(somSelf) | CLSSTYLE_NEVERPRINT );
}
```

wpclsQueryStyle - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpclsQueryTitle

wpclsQueryTitle - Syntax

This method is called to allow the class object to specify the default title for its instances.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;  /*  Pointer to the WPObject class object. */
PSZ             pTitle;   /*  Pointer to the default title for objects of this class. */

pTitle = _wpclsQueryTitle(somSelf);
```

wpclsQueryTitle Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsQueryTitle Return Value - pTitle

pTitle (PSZ) - returns
Pointer to the default title for objects of this class.

wpclsQueryTitle - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

pTitle (PSZ) - returns
Pointer to the default title for objects of this class.

wpclsQueryTitle - Remarks

The title is used as the default for new instances of this class. In addition, the title is used to describe the class in facilities such as Find, Include, Details, and Sort. The title can be loaded on wpclsInitData and freed on wpclsUnInitData.

wpclsQueryTitle - Usage

This method can be called at any time in order to determine the default title for instances of this class.

wpclsQueryTitle - How to Override

All classes should override this method, so that new objects and their classes always have a sensible default title.

wpclsQueryTitle - Related Methods

Related Methods

- wpQueryTitle
- wpclsQueryTitle
- wpSetTitle

wpclsQueryTitle - Example Code

This example overrides this method to provide another title.

```
SOM_Scope PSZ   SOMLINK SampleM_wpclsQueryTitle(M_Sample *somSelf)
{
    /* M_SampleData *somThis = M_SampleGetData(somSelf); */
    M_SampleMethodDebug("M_Sample","SampleM_wpclsQueryTitle");

    return( "My WPS Object" );                /* Return a new title */
}
```

wpclsQueryTitle - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpclsRemoveObjects

wpclsRemoveObjects - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This class method removes all specified objects from a given container and the in-use list.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;       /*  Pointer to the WPObject class object. */
HWND            hwndCnr;       /*  Handle to the container object. */
PVOID          *pRecordArray;  /*  Pointer to an array of MINIRECORDCORE objects that are to be removed. */
ULONG           NumRecords;    /*  Number of records in pRecordArray. */
BOOL            RemoveAll;     /*  Flag indicating whether to remove all records. */
BOOL            rc;            /*  Success indicator. */

rc = _wpclsRemoveObjects(somSelf, hwndCnr,
       pRecordArray, NumRecords, RemoveAll);
```

wpclsRemoveObjects Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsRemoveObjects Parameter - hwndCnr

hwndCnr (HWND) - input
Handle to the container object.

wpclsRemoveObjects Parameter - pRecordArray

pRecordArray (PVOID *) - input
Pointer to an array of MINIRECORDCORE objects that are to be removed.

wpclsRemoveObjects Parameter - NumRecords

NumRecords (ULONG) - input
Number of records in *pRecordArray* .

wpclsRemoveObjects Parameter - RemoveAll

RemoveAll (BOOL) - input
Flag indicating whether to remove all records.

Possible values are described in the following list:

TRUE
All records in the specified container are to be removed, and the in-use list for each object is to be cleaned up.

FALSE
Only records specified in *pRecordArray* are to be removed, and the in-use list is to be cleaned up for the same objects.

wpclsRemoveObjects Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.

FALSE
Error occurred.

wpclsRemoveObjects - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

hwndCnr (HWND) - input
Handle to the container object.

pRecordArray (PVOID *) - input
Pointer to an array of MINIRECORDCORE objects that are to be removed.

NumRecords (ULONG) - input
Number of records in *pRecordArray* .

RemoveAll (BOOL) - input
Flag indicating whether to remove all records.

Possible values are described in the following list:

TRUE
All records in the specified container are to be removed, and the in-use list for each object is to be cleaned up.

FALSE
Only records specified in *pRecordArray* are to be removed, and the in-use list is to be cleaned up for the same objects.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.

FALSE
Error occurred.

wpclsRemoveObjects - Remarks

This method provides rapid removal of multiple objects from a container at one time. wpCnrRemoveObject performs the same function as this method, but it operates only on one object at a time.

wpclsRemoveObjects - Usage

This method can be called at any time to remove multiple objects from the container.

wpclsRemoveObjects - How to Override

This method is generally not overridden.

wpclsRemoveObjects - Related Methods

Related Methods

- wpCnrInsertObject
- wpCnrRemoveObject
- wpclsInsertMultipleObjects

wpclsRemoveObjects - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpclsSetError

wpclsSetError - Syntax

This class method sets the current error code within a class object.

```
#define INCL_WINWORKPLACE
```

```
#include <os2.h>

M_WPObject     *somSelf;    /*  Pointer to the WPObject class object. */
ULONG           ulErrorId;  /*  Error code. */
BOOL            rc;         /*  Success indicator. */

rc = _wpclsSetError(somSelf, ulErrorId);
```

wpclsSetError Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsSetError Parameter - ulErrorId

ulErrorId (ULONG) - input
Error code.

wpclsSetError Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	The error was successfully stored.
FALSE	The error was not successfully stored.

wpclsSetError - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

ulErrorId (ULONG) - input
Error code.

rc (BOOL) - returns
Success indicator.

TRUE	The error was successfully stored.
FALSE	The error was not successfully stored.

wpclsSetError - Remarks

This method can be used when writing class methods for workplace objects that return BOOLEAN values. When a given class method fails, the class method can log its error code in the class object so that the caller can later retrieve it using the wpclsQueryError class method. The function is analogous to the *WinSetErrorInfo* function call that is used by Presentation Manager functions to log their error return codes.

wpclsSetError - Usage

It is recommended that this method be called only by all class methods when they do not execute successfully and nowhere else.

wpclsSetError - How to Override

Never override this class method.

wpclsSetError - Related Methods

Related Methods

- wpclsQueryError

wpclsSetError - Example Code

This example finds all occurrences of an object type in a folder.

```
 VOID  clsFindEverybody(SOMAny *self)
{
   BOOL      rc;                 /* Return from methods */
   HFIND     hFind       = 0;    /* Handle for repeated finds */
   CLASS     Class[2];           /* List of classes to look for */
   OBJECT    Object[10];         /* Objects returned from search */
   ULONG     ulCount     = 10;   /* Count of objects to find / or found */
   WPFolder  *folder     = 0;    /* Pointer to folder for search */
   ULONG     errorid     = 0;    /* Error returned from previous search */
   CHAR      sz[CCHMAXPATH];     /* Title string to look for */
   PSZ       title = sz;

   Class[0] = _WPFolder;
   Class[1] = NULL;


   /* Get the pointer to the desktop (since that is where we are looking) */

   folder = _wpclsQueryFolder( _WPFolder, "<WP_DESKTOP>", TRUE );

   _wpclsSetError( _WPObject, 0 );      /* Reset the error indicators */
   rc = 0;

         /* Attempt to find the first 10 objects matching our criteria */

   ulCount = 10;
   rc =_wpclsFindObjectFirst(_WPObject,     /* Class self pointer */
                             Class,         /* Pointer to array of class objects */
                             &hFind,        /* Handle for wpclsFindObjectNext */
                             NULL,          /* Find EVERYTHING */
                             folder,        /* Pointer to folder of interest */
                             TRUE,          /* Search all folders in tree */
                             NULL,          /* No extended criteria */
```

```
                              Object,        /* Array of object pointers */
                              &ulCount );    /* Number requested and number found */

   errorid = _wpclsQueryError( _WPObject);  /* Get errorid (may still be OK) */


   if ((!rc && (errorid == WPERR_BUFFER_OVERFLOW)) || rc) {

               /***** Process objects found here ******/

   } else {
      somPrintf(" _wpclsFindObjectFirst failed, errorid=%u\n", errorid );
      return;
   } /* endif */

   do {

      _wpclsSetError( _WPObject, 0 );    /* Reset error indicators */
      rc = 0;

             /* Look for next 10 objects */

      rc =_wpclsFindObjectNext(_WPObject,       /* Self */
                               hFind,           /* Handle from previous Find */
                               Object,          /* Objects found */
                               &ulCount );      /* Number to look for. On
                                                   return, number found */

      errorid = _wpclsQueryError( _WPObject );

      if ((!rc && (errorid == WPERR_BUFFER_OVERFLOW)) || rc) {


              /***** Process objects found here ******/

      } else {
            somPrintf("_wpclsFindObjectNext failed, errorid=%u\n", errorid);
            return;
      } /* endif */

   } while ( errorid == WPERR_BUFFER_OVERFLOW ); /* enddo */

   _wpclsSetError( _WPObject, 0 );    /* Reset error indicators */
   rc = 0;

               /* Indicate find complete */

   rc =_wpclsFindObjectEnd( _WPObject, hFind );

   if (!rc) {
     errorid = _wpclsQueryError( _WPObject );
     somPrintf( "_wpclsFindObjectEnd failed, errorid=%u\n", errorid );
     return;
   }

   return;
}
```

wpclsSetError - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpclsSetIcon

wpclsSetIcon - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This class method sets the icon handle for a given object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;      /*  Pointer to the WPObject class object. */
HPOINTER        hptrNewIcon;  /*  Icon handle to be set for the object. */
BOOL            rc;           /*  Success indicator. */

rc = _wpclsSetIcon(somSelf, hptrNewIcon);
```

wpclsSetIcon Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsSetIcon Parameter - hptrNewIcon

hptrNewIcon (HPOINTER) - input
Icon handle to be set for the object.

wpclsSetIcon Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpclsSetIcon - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

hptrNewIcon (HPOINTER) - input
Icon handle to be set for the object.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpclsSetIcon - Remarks

The class default icon can be loaded using wpclsInitData and freed using wpclsUnInitData.

wpclsSetIcon - Usage

This method can be called at any time to set the object's default icon.

wpclsSetIcon - How to Override

This method is generally not overridden.

wpclsSetIcon - Related Methods

Related Methods

- wpclsQueryIcon
- wpQueryIcon
- wpQueryIconData
- wpSetIcon
- wpSetIconData
- wpclsQueryIcon
- wpclsQueryIconData

wpclsSetIcon - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage

Related Methods
Glossary

wpclsSetIconData

wpclsSetIconData - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This class method sets the icon information for a given object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;    /*  Pointer to the WPObject class object. */
PICONINFO       pIconInfo;  /*  Pointer to the icon information structure. */
BOOL            rc;         /*  Success indicator. */

rc = _wpclsSetIconData(somSelf, pIconInfo);
```

wpclsSetIconData Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsSetIconData Parameter - pIconInfo

pIconInfo (PICONINFO) - input
Pointer to the icon information structure.

wpclsSetIconData Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.

FALSE
Error occurred.

wpclsSetIconData - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

pIconInfo (PICONINFO) - input
Pointer to the icon information structure.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpclsSetIconData - Usage

This method can be called at any time to set icon information.

wpclsSetIconData - How to Override

wpclsSetIconData - Related Methods

Related Methods

- wpclsQueryIcon
- wpQueryIcon
- wpQueryIconData
- wpSetIcon
- wpSetIconData
- wpclsQueryIconData

wpclsSetIconData - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpclsSetSetting

wpclsSetSetting - Syntax

This method sets the class object settings in the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPAbstract     *somSelf;      /*  Pointer to the WPAbstract class object. */
PSZ               pszSetting;   /*  Name of the setting that is to be set. */
PVOID             pValue;       /*  Buffer which contains setting values. */
BOOL              rc;           /*  Success indicator. */

rc = _wpclsSetSetting(somSelf, pszSetting,
       pValue);
```

wpclsSetSetting Parameter - somSelf

somSelf (M_WPAbstract *) - input
Pointer to the WPAbstract class object.

wpclsSetSetting Parameter - pszSetting

pszSetting (PSZ) - input
Name of the setting that is to be set.

wpclsSetSetting Parameter - pValue

pValue (PVOID) - input
Buffer which contains setting values.

wpclsSetSetting Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpclsSetSetting - Parameters

somSelf (M_WPAbstract *) - input
Pointer to the WPAbstract class object.

pszSetting (PSZ) - input
Name of the setting that is to be set.

pValue (PVOID) - input
Buffer which contains setting values.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpclsSetSetting - Remarks

This is a virtual method and must be overridden by a subclass. Each object subclass documents the possible values for *pszSetting* and *pValue* . These settings and values are specific for each subclass. Note that all settings have safe defaults, so it is never necessary to pass unnecessary settings to an object.

For a list of settings and their associated values, see the individual classes.

wpclsSetSetting - Usage

This method should never be called. It is a virtual method and must be overridden by a subclass.

wpclsSetSetting - How to Override

This is a virtual method and must be overridden by a subclass.

wpclsSetSetting - Related Methods

Related Methods

- wpclsSetSetting

wpclsSetSetting - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpclsSetSettingsPageSize

wpclsSetSettingsPageSize - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This class method returns the default Settings page size.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;  /*  Pointer to the WPObject class object. */
PSIZEL          pSizel;   /*  Pointer to the structure containing the default width and height of the Settings p
BOOL            rc;       /*  Success indicator. */

rc = _wpclsSetSettingsPageSize(somSelf, pSizel);
```

wpclsSetSettingsPageSize Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsSetSettingsPageSize Parameter - pSizel

pSizel (PSIZEL) - input
Pointer to the structure containing the default width and height of the Settings page.

wpclsSetSettingsPageSize Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpclsSetSettingsPageSize - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

pSizel (PSIZEL) - input
Pointer to the structure containing the default width and height of the Settings page.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpclsSetSettingsPageSize - How to Override

This method is generally not overridden.

wpclsSetSettingsPageSize - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Glossary

wpclsUnInitData

wpclsUnInitData - Syntax

This method is called to allow the class object to free allocated resources.

```
#define INCL_WINWORKPLACE
#include <os2.h>

M_WPObject     *somSelf;  /*  Pointer to the WPObject class object. */

wpclsUnInitData(somSelf);
```

wpclsUnInitData Parameter - somSelf

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

wpclsUnInitData - Return Value

There is no return value for this method.

wpclsUnInitData - Parameters

somSelf (M_WPObject *) - input
Pointer to the WPObject class object.

There is no return value for this method.

wpclsUnInitData - Usage

This method is generally called only by the system when the class object is made dormant. The class object is made dormant when the last instance of this class is made dormant.

wpclsUnInitData - How to Override

Any class that overrides the wpclsInitData method to allocate resources for its metaclass instance variables should override this method to deallocate those resources. It is essential to pass this method onto the parent class object after performing override processing.

wpclsUnInitData - Related Methods

Related Methods

- wpclsInitData

wpclsUnInitData - Example Code

This example overrides the method to free any data associated with the class.

```
SOM_Scope void SOMLINK BearsM_wpclsUnInitData(M_Bears *somSelf)
{
    APIRET  rc  =  NO_ERROR;            /* Return code from Dos API */

    M_BearsData *somThis = M_BearsGetData(somSelf);
    M_BearsMethodDebug("M_Bears","myfM_wpclsUnInitData");

    rc = DosFreeMem( _pBearClassData );

    parent_wpclsUnInitData(somSelf);

}
```

wpclsUnInitData - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

Workplace Instance Methods

This chapter contains an alphabetic listing of the Workplace instance methods. Instance methods are methods that are valid for a specific object.

Note: In scanning the header files, you may find the names of Workplace methods that are not documented in this material. There are several places this might happen: in Class Data Structures, Class Method Tables, or Release Order Lists. Methods that are listed in one of these locations, but not documented, are private methods. They are exposed in header files due to the way the SOM compiler generates the header files.

wpAddDDEPage

wpAddDDEPage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to add the DDE Details page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPWinConfig     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND             hwndNotebook;  /*  Handle to the Settings notebook. */
ULONG            ulPageid;      /*  Identifier for the inserted page. */

ulPageid = _wpAddDDEPage(somSelf, hwndNotebook);
```

wpAddDDEPage Parameter - somSelf

somSelf (WPWinConfig *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPWinConfig.

wpAddDDEPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Handle to the Settings notebook.

wpAddDDEPage Return Value - ulPageid

ulPageid (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the DDE details page was not added to the Settings notebook.

wpAddDDEPage - Parameters

somSelf (WPWinConfig *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPWinConfig.

hwndNotebook (HWND) - input
Handle to the Settings notebook.

ulPageid (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the DDE details page was not added to the Settings notebook.

wpAddDDEPage - Usage

This method can be called at any time.

wpAddDDEPage - How to Override

This method should always be overridden to replace or remove the Dynamic Data Exchange (DDE) Details page from an object which is a descendant of WPWinConfig. An override of this method does not call the parent.

wpAddDDEPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddDDEPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override

Usage
Related Methods
Glossary

wpAddDMQSDisplayTypePage

wpAddDMQSDisplayTypePage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to add the DMQS Display Type page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSystem     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Window handle to the Settings notebook. */
ULONG         ulPageId;      /*  Identifier for the inserted page. */

ulPageId = _wpAddDMQSDisplayTypePage(somSelf,
             hwndNotebook);
```

wpAddDMQSDisplayTypePage Parameter - somSelf

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

wpAddDMQSDisplayTypePage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Window handle to the Settings notebook.

wpAddDMQSDisplayTypePage Return Value - ulPageId

ulPageId (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the DMQS Display Type page was not added to the Settings notebook.

wpAddDMQSDisplayTypePage - Parameters

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

hwndNotebook (HWND) - input
Window handle to the Settings notebook.

ulPageId (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the DMQS Display Type page was not added to the Settings notebook.

wpAddDMQSDisplayTypePage - How to Override

This method should always be overridden to replace or remove the DMQS Display Type page from an object which is a descendant of WPSystem. An override of this method does not call the parent.

wpAddDMQSDisplayTypePage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddDMQSDisplayTypePage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpAddBecomePage

wpAddBecomePage - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method adds the standard type page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDataFile     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND            hwndNotebook;  /*  Settings notebook handle. */
ULONG           rc;            /*  Page indicator. */

rc = _wpAddBecomePage(somSelf, hwndNotebook);
```

wpAddBecomePage Parameter - somSelf

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

wpAddBecomePage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddBecomePage Return Value - rc

rc (ULONG) - returns
Page indicator.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddBecomePage - Parameters

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns

Page indicator.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddBecomePage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddBecomePage - How to Override

This method should always be overridden in order to replace or remove the Become page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddBecomePage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddBecomePage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddBecomePage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddBecomePage");

/*  return (parent_wpAddBecomePage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddBecomePage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods

wpAddClockAlarmPage

wpAddClockAlarmPage - Syntax

This instance method is called to allow the object to add the *Alarm* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPClock     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Settings notebook handle. */
ULONG        rc;            /*  Page identifier. */

rc = _wpAddClockAlarmPage(somSelf, hwndNotebook);
```

wpAddClockAlarmPage Parameter - somSelf

somSelf (WPClock *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPClock.

wpAddClockAlarmPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddClockAlarmPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddClockAlarmPage - Parameters

somSelf (WPClock *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPClock.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddClockAlarmPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddClockAlarmPage - How to Override

This method should always be overridden in order to remove or replace the *Alarm* page from the object's Settings notebook.

To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddClockAlarmPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddClockAlarmPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddClockAlarmPage(MyObj *somSelf,
                HWND hwndNotebook)

{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddClockAlarmPage");

/*  return (parent_wpAddClockAlarmPage(somSelf,hwndNotebook)); */
    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddClockAlarmPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddClockDateTimePage

wpAddClockDateTimePage - Syntax

This instance method is called to allow the object to add the *Date/Time* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPClock     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Settings notebook handle. */
ULONG        rc;            /*  Page identifier. */

rc = _wpAddClockDateTimePage(somSelf, hwndNotebook);
```

wpAddClockDateTimePage Parameter - somSelf

somSelf (WPClock *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPClock.

wpAddClockDateTimePage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddClockDateTimePage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddClockDateTimePage - Parameters

somSelf (WPClock *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPClock.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddClockDateTimePage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddClockDateTimePage - How to Override

This method should always be overridden in order to replace or remove the *Date/Time* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddClockDateTimePage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddClockDateTimePage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddClockDateTimePage(MyObj *somSelf,
                HWND hwndNotebook)

{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddClockDateTimePage");

/*  return (parent_wpAddClockDateTimePage(somSelf,hwndNotebook)); */
    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddClockDateTimePage - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddClockView1Page

wpAddClockView1Page - Syntax

This instance method is called to allow the object to add the *View 1* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPClock     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Settings notebook handle. */
ULONG        rc;            /*  Page identifier. */

rc = _wpAddClockView1Page(somSelf, hwndNotebook);
```

wpAddClockView1Page Parameter - somSelf

somSelf (WPClock *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPClock.

wpAddClockView1Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddClockView1Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddClockView1Page - Parameters

somSelf (WPClock *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPClock.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddClockView1Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddClockView1Page - How to Override

This method should always be overridden in order to replace or remove the *View 1* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddClockView1Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddClockView1Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddClockView1Page(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddClockView1Page");

/*  return (parent_wpAddClockView1Page(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddClockView1Page - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddClockView2Page

wpAddClockView2Page - Syntax

This instance method is called to allow the object to add the *View 2* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPClock     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Settings notebook handle. */
```

```
ULONG        rc;             /*  Page identifier. */

rc = _wpAddClockView2Page(somSelf, hwndNotebook);
```

wpAddClockView2Page Parameter - somSelf

somSelf (WPClock *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPClock.

wpAddClockView2Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddClockView2Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddClockView2Page - Parameters

somSelf (WPClock *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPClock.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddClockView2Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddClockView2Page - How to Override

This method should always be overridden in order to replace or remove the *View 2* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddClockView2Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddClockView2Page - Example Code

This example overrrides this method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddClockView2Page(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddClockView2Page");

/*  return (parent_wpAddClockView2Page(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddClockView2Page - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddCountryDatePage

wpAddCountryDatePage - Syntax

This instance method is called to allow the object to add the *Date* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPCountry     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND           hwndNotebook;  /*  Settings notebook handle. */
ULONG          rc;            /*  Page identifier. */

rc = _wpAddCountryDatePage(somSelf, hwndNotebook);
```

wpAddCountryDatePage Parameter - somSelf

somSelf (WPCountry *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPCountry.

wpAddCountryDatePage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddCountryDatePage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddCountryDatePage - Parameters

somSelf (WPCountry *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPCountry.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddCountryDatePage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddCountryDatePage - How to Override

This method should always be overridden in order to replace or remove the *Date* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddCountryDatePage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddCountryDatePage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddCountryDatePage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddCountryDatePage");

/*  return (parent_wpAddCountryDatePage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddCountryDatePage - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddCountryNumbersPage

wpAddCountryNumbersPage - Syntax

This instance method is called to allow the object to add the *Numbers* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPCountry     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND           hwndNotebook;  /*  Settings notebook handle. */
ULONG          rc;            /*  Page identifier. */

rc = _wpAddCountryNumbersPage(somSelf, hwndNotebook);
```

wpAddCountryNumbersPage Parameter - somSelf

somSelf (WPCountry *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPCountry.

wpAddCountryNumbersPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddCountryNumbersPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddCountryNumbersPage - Parameters

somSelf (WPCountry *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPCountry.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddCountryNumbersPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddCountryNumbersPage - How to Override

This method should always be overridden in order to replace or remove the *Numbers* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddCountryNumbersPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddCountryNumbersPage - Example Code

This examples overrides this method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddCountryNumbersPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
```

```
    MyObjMethodDebug("MyObj","MyObj_wpAddCountryNumbersPage");

/*  return (parent_wpAddCountryNumbersPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddCountryNumbersPage - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddCountryPage

wpAddCountryPage - Syntax

This instance method is called to allow the object to add the *Country* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPCountry     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND           hwndNotebook;  /*  Settings notebook handle. */
ULONG          rc;            /*  Page identifier. */

rc = _wpAddCountryPage(somSelf, hwndNotebook);
```

wpAddCountryPage Parameter - somSelf

somSelf (WPCountry *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPCountry.

wpAddCountryPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddCountryPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddCountryPage - Parameters

somSelf (WPCountry *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPCountry.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddCountryPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddCountryPage - How to Override

This method should always be overridden in order to replace or remove the *Country* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddCountryPage - Related Methods

Related Methods

- wpAddSettingsPages

- wpInsertSettingsPage

wpAddCountryPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddCountryPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddCountryPage");

/*  return (parent_wpAddCountryPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddCountryPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddCountryTimePage

wpAddCountryTimePage - Syntax

This instance method is called to allow the object to add the *Time* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPCountry     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND           hwndNotebook;  /*  Settings notebook handle. */
ULONG          rc;            /*  Page identifier. */

rc = _wpAddCountryTimePage(somSelf, hwndNotebook);
```

wpAddCountryTimePage Parameter - somSelf

somSelf (WPCountry *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPCountry.

wpAddCountryTimePage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddCountryTimePage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddCountryTimePage - Parameters

somSelf (WPCountry *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPCountry.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddCountryTimePage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddCountryTimePage - How to Override

This method should always be overridden in order to replace or remove the *Time* page from the object's Settings notebook. To remove the

page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddCountryTimePage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddCountryTimePage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddCountryTimePage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddCountryTimePage");

/*  return (parent_wpAddCountryTimePage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddCountryTimePage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddDesktopArcRest1Page

wpAddDesktopArcRest1Page - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to allow the object to add the *Archive* page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND           hwndNotebook;  /*  Settings notebook handle. */
ULONG          rc;            /*  Page identifier. */

rc = _wpAddDesktopArcRest1Page(somSelf, hwndNotebook);
```

wpAddDesktopArcRest1Page Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpAddDesktopArcRest1Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddDesktopArcRest1Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0
Error occurred
PageId
Identifier for the inserted page.

wpAddDesktopArcRest1Page - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0

	Error occurred
PageId	Identifier for the inserted page.

wpAddDesktopArcRest1Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddDesktopArcRest1Page - How to Override

This method should always be overridden to replace or remove the *Archive* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method must return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method must call the wpInsertSettingsPage method without calling the parent method.

wpAddDesktopArcRest1Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddDesktopArcRest1Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddDesktopArcRest1Page(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddDesktopArcRest1Page");

/*  return (parent_wpAddDesktopArcRest1Page(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddDesktopArcRest1Page - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods

wpAddDesktopDefDT1Page

wpAddDesktopDefDT1Page - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to allow the object to add the *Desktop* page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND           hwndNotebook;  /*  Settings notebook handle. */
ULONG          rc;            /*  Page identifier. */

rc = _wpAddDesktopDefDT1Page(somSelf, hwndNotebook);
```

wpAddDesktopDefDT1Page Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpAddDesktopDefDT1Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddDesktopDefDT1Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0
Error occurred

PageId
Identifier for the inserted page.

wpAddDesktopDefDT1Page - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0
Error occurred
PageId
Identifier for the inserted page.

wpAddDesktopDefDT1Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddDesktopDefDT1Page - How to Override

This method should always be overridden to replace or remove the *Desktop* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddDesktopDefDT1Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddDesktopDefDT1Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddDesktopDefDT1Page(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddDesktopDefDT1Page");
```

```
/*  return (parent_wpAddDesktopDefDT1Page(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddDesktopDefDT1Page - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddDesktopLockup1Page

wpAddDesktopLockup1Page - Syntax

This instance method is called to allow the object to add the *Lockup 1* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND           hwndNotebook;  /*  Settings notebook handle. */
ULONG          rc;            /*  Page identifier. */

rc = _wpAddDesktopLockup1Page(somSelf, hwndNotebook);
```

wpAddDesktopLockup1Page Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpAddDesktopLockup1Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddDesktopLockup1Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddDesktopLockup1Page - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddDesktopLockup1Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddDesktopLockup1Page - How to Override

This method should always be overridden in order to replace or remove the *Lockup 1* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddDesktopLockup1Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddDesktopLockup1Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddDesktopLockup1Page(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddDesktopLockup1Page");

/*  return (parent_wpAddDesktopLockup1Page(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddDesktopLockup1Page - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddDesktopLockup2Page

wpAddDesktopLockup2Page - Syntax

This instance method is called to allow the object to add the *Lockup 2* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND           hwndNotebook;  /*  Settings notebook handle. */
ULONG          rc;            /*  Page identifier. */

rc = _wpAddDesktopLockup2Page(somSelf, hwndNotebook);
```

wpAddDesktopLockup2Page Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpAddDesktopLockup2Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddDesktopLockup2Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddDesktopLockup2Page - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddDesktopLockup2Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddDesktopLockup2Page - How to Override

This method should always be overridden in order to replace or remove the *Lockup 2* page from the object's Settings notebook. To remove

the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddDesktopLockup2Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddDesktopLockup2Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddDesktopLockup2Page(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddDesktopLockup2Page");

/*  return (parent_wpAddDesktopLockup2Page(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddDesktopLockup2Page - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddDesktopLockup3Page

wpAddDesktopLockup3Page - Syntax

This instance method is called to allow the object to add the *Lockup 3* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND           hwndNotebook;  /*  Settings notebook handle. */
ULONG          rc;            /*  Page identifier. */

rc = _wpAddDesktopLockup3Page(somSelf, hwndNotebook);
```

wpAddDesktopLockup3Page Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpAddDesktopLockup3Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddDesktopLockup3Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddDesktopLockup3Page - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddDesktopLockup3Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddDesktopLockup3Page - How to Override

This method should always be overridden in order to replace or remove the *Lockup 3* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddDesktopLockup3Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddDesktopLockup3Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddDesktopLockup3Page(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddDesktopLockup3Page");

/*  return (parent_wpAddDesktopLockup3Page(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddDesktopLockup3Page - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddDeviceDevice1Page

wpAddDeviceDevice1Page - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method is called to allow an object to add the first Device page to the settings notebook.

```
#define INCL_WORKPLACE
#include <os2.h>

WPDevice     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddDeviceDevice1Page(somSelf, hwndNotebook);
```

wpAddDeviceDevice1Page Parameter - somSelf

somSelf (WPDevice *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDevice.

wpAddDeviceDevice1Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddDeviceDevice1Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddDeviceDevice1Page - Parameters

somSelf (WPDevice *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDevice.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddDeviceDevice1Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddDeviceDevice1Page - How to Override

This method should always be overridden to replace or remove the first Device page from the object's Settings notebook. To remove the page from the settings notebook, the override method must return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method must call the wpInsertSettingsPage wpAddSettingsPages method without calling the parent method.

wpAddDeviceDevice1Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddDeviceDevice1Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG SOMLINK MyObj_wpAddDeviceDevice1Page(MyObj *somSelf,
                                                     HWND hwndNotebook)

{
  /* MyObjData *somThis = MyObjGetData(somSelf); */
  MyObjMethodDebug("MyObj","MyObj_wpAddDeviceDevice1Page");

  /* return(parent_wpAddDeviceDevice1Page(somSelf,hwndNotebook)); */

  return(SETTINGS_PAGE_REMOVED);
}
```

wpAddDeviceDevice1Page - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddDeviceDevice2Page

wpAddDeviceDevice2Page - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method is called to allow an object to add the second Device page to the settings notebook.

```
#define INCL_WORKPLACE
#include <os2.h>

WPDevice     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddDeviceDevice2Page(somSelf, hwndNotebook);
```

wpAddDeviceDevice2Page Parameter - somSelf

somSelf (WPDevice *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDevice.

wpAddDeviceDevice2Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddDeviceDevice2Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddDeviceDevice2Page - Parameters

somSelf (WPDevice *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDevice.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddDeviceDevice2Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddDeviceDevice2Page - How to Override

This method should always be overridden to replace or remove the second Device page from the object's Settings notebook. To remove the page from the settings notebook, the override method must return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method must call the wpAddSettingsPages method without calling the parent method.

wpAddDeviceDevice2Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddDeviceDevice2Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG SOMLINK MyObj_wpAddDeviceDevice2Page(MyObj *somSelf,
                                                     HWND hwndNotebook)

{
  /* MyObjData *somThis = MyObjGetData(somSelf); */
  MyObjMethodDebug("MyObj","MyObj_wpAddDeviceDevice2Page");

  /* return(parent_wpAddDeviceDevice2Page(somSelf,hwndNotebook)); */

  return(SETTINGS_PAGE_REMOVED);
}
```

wpAddDeviceDevice2Page - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddDeviceDMA1Page

wpAddDeviceDMA1Page - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method is called to allow an object to add the DMA Channel page to the settings notebook.

```
#define INCL_WORKPLACE
#include <os2.h>

WPDevice     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddDeviceDMA1Page(somSelf, hwndNotebook);
```

wpAddDeviceDMA1Page Parameter - somSelf

somSelf (WPDevice *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDevice.

wpAddDeviceDMA1Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddDeviceDMA1Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddDeviceDMA1Page - Parameters

somSelf (WPDevice *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDevice.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddDeviceDMA1Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddDeviceDMA1Page - How to Override

This method should always be overridden to replace or remove the DMA Channel page from the object's Settings notebook. To remove the page from the settings notebook, the override method must return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method must call the wpAddSettingsPages method without calling the parent method.

wpAddDeviceDMA1Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddDeviceDMA1Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG SOMLINK MyObj_wpAddDeviceDMA1Page(MyObj *somSelf,
                                                  HWND hwndNotebook)

{
  /* MyObjData *somThis = MyObjGetData(somSelf); */
  MyObjMethodDebug("MyObj","MyObj_wpAddDeviceDMA1Page");

  /* return(parent_wpAddDeviceDMA1Page(somSelf,hwndNotebook)); */

  return(SETTINGS_PAGE_REMOVED);
}
```

wpAddDeviceDMA1Page - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddDeviceIO1Page

wpAddDeviceIO1Page - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method is called to allow an object to add the I/O Ports page to the settings notebook.

```
#define INCL_WORKPLACE
#include <os2.h>

WPDevice     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */
```

```
rc = _wpAddDeviceIO1Page(somSelf, hwndNotebook);
```

wpAddDeviceIO1Page Parameter - somSelf

somSelf (WPDevice *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDevice.

wpAddDeviceIO1Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddDeviceIO1Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddDeviceIO1Page - Parameters

somSelf (WPDevice *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDevice.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddDeviceIO1Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddDeviceIO1Page - How to Override

This method should always be overridden to replace or remove the I/O Ports page from the object's Settings notebook. To remove the page from the settings notebook, the override method must return SETINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method must call the wpAddSettingsPages method without calling the parent method.

wpAddDeviceIO1Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddDeviceIO1Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG SOMLINK MyObj_wpAddDeviceIO1Page(MyObj *somSelf,
                                                 HWND hwndNotebook)

{
  /* MyObjData *somThis = MyObjGetData(somSelf); */
  MyObjMethodDebug("MyObj","MyObj_wpAddDeviceIO1Page");

  /* return(parent_wpAddDeviceIO1Page(somSelf,hwndNotebook)); */

  return(SETTINGS_PAGE_REMOVED);
}
```

wpAddDeviceIO1Page - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddDeviceIRQ1Page

wpAddDeviceIRQ1Page - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method is called to allow an object to add the IRQ Level page to the settings notebook.

```
#define INCL_WORKPLACE
#include <os2.h>

WPDevice     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddDeviceIRQ1Page(somSelf, hwndNotebook);
```

wpAddDeviceIRQ1Page Parameter - somSelf

somSelf (WPDevice *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDevice.

wpAddDeviceIRQ1Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddDeviceIRQ1Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddDeviceIRQ1Page - Parameters

somSelf (WPDevice *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDevice.

hwndNotebook (HWND) - input

Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddDeviceIRQ1Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddDeviceIRQ1Page - How to Override

This method should always be overridden to replace or remove the IRQ Level page from the object's Settings notebook. To remove the page from the settings notebook, the override method must return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method must call the wpAddSettingsPages method without calling the parent method.

wpAddDeviceIRQ1Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddDeviceIRQ1Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG SOMLINK MyObj_wpAddDeviceIRQ1Page(MyObj *somSelf,
                                                  HWND hwndNotebook)

{
  /* MyObjData *somThis = MyObjGetData(somSelf); */
  MyObjMethodDebug("MyObj","MyObj_wpAddDeviceIRQ1Page");

  /* return(parent_wpAddDeviceIRQ1Page(somSelf,hwndNotebook)); */

  return(SETTINGS_PAGE_REMOVED);
}
```

wpAddDeviceIRQ1Page - Topics

Select an item:
Syntax
Parameters
Returns

How to Override
Usage
Example Code
Related Methods
Glossary

wpAddDeviceMemory1Page

wpAddDeviceMemory1Page - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method is called to allow an object to add the Memory page to the settings notebook.

```
#define INCL_WORKPLACE
#include <os2.h>

WPDevice     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddDeviceMemory1Page(somSelf, hwndNotebook);
```

wpAddDeviceMemory1Page Parameter - somSelf

somSelf (WPDevice *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDevice.

wpAddDeviceMemory1Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddDeviceMemory1Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddDeviceMemory1Page - Parameters

somSelf (WPDevice *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDevice.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddDeviceMemory1Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddDeviceMemory1Page - How to Override

This method should always be overridden to replace or remove the Memory page from the object's Settings notebook. To remove the page from the settings notebook, the override method must return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method must call the wpAddSettingsPages method without calling the parent method.

wpAddDeviceMemory1Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddDeviceMemory1Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG SOMLINK MyObj_wpAddDeviceMemory1Page(MyObj *somSelf,
                                                     HWND hwndNotebook)

{
  /* MyObjData *somThis = MyObjGetData(somSelf); */
  MyObjMethodDebug("MyObj","MyObj_wpAddDeviceMemory1Page");
```

```
  /* return(parent_wpAddDeviceMemory1Page(somSelf,hwndNotebook)); */

  return(SETTINGS_PAGE_REMOVED);
}
```

wpAddDeviceMemory1Page - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddDiskDetailsPage

wpAddDiskDetailsPage - Syntax

This instance method is called to allow the object to add the *Details* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDisk     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND        hwndNotebook;  /*  Settings notebook handle. */
ULONG       rc;            /*  Page identifier. */

rc = _wpAddDiskDetailsPage(somSelf, hwndNotebook);
```

wpAddDiskDetailsPage Parameter - somSelf

somSelf (WPDisk *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDisk.

wpAddDiskDetailsPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddDiskDetailsPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddDiskDetailsPage - Parameters

somSelf (WPDisk *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDisk.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddDiskDetailsPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddDiskDetailsPage - How to Override

This method should always be overridden in order to replace or remove the *Details* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddDiskDetailsPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddDiskDetailsPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddDiskDetailsPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddDiskDetailsPage");

/*  return (parent_wpAddDiskDetailsPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddDiskDetailsPage - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddFile1Page

wpAddFile1Page - Syntax

This instance method is called to allow the object to add the *File 1* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND              hwndNotebook;  /*  Settings notebook handle. */
ULONG             rc;            /*  Page identifier. */

rc = _wpAddFile1Page(somSelf, hwndNotebook);
```

wpAddFile1Page Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpAddFile1Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddFile1Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFile1Page - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFile1Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddFile1Page - How to Override

This method should always be overridden in order to replace or remove the *File 1* page from the object's Settings notebook. To remove the

page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddFile1Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddFile1Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddFile1Page(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddFile1Page");

/*  return (parent_wpAddFile1Page(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddFile1Page - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddFile2Page

wpAddFile2Page - Syntax

This instance method is called to allow the object to add the *File 2* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND              hwndNotebook;  /*  Settings notebook handle. */
ULONG             rc;            /*  Page identifier. */

rc = _wpAddFile2Page(somSelf, hwndNotebook);
```

wpAddFile2Page Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpAddFile2Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddFile2Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFile2Page - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFile2Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddFile2Page - How to Override

This method should always be overridden in order to replace or remove the *File 2* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddFile2Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddFile2Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddFile2Page(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddFile2Page");

/*  return (parent_wpAddFile2Page(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddFile2Page - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddFile3Page

wpAddFile3Page - Syntax

This instance method is called to allow the object to add the *File 3* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND              hwndNotebook;  /*  Settings notebook handle. */
ULONG             rc;            /*  Page identifier. */

rc = _wpAddFile3Page(somSelf, hwndNotebook);
```

wpAddFile3Page Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpAddFile3Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddFile3Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFile3Page - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFile3Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddFile3Page - How to Override

This method should always be overridden in order to replace or remove the *File 3* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddFile3Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddFile3Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddFile3Page(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddFile3Page");

/*  return (parent_wpAddFile3Page(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddFile3Page - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddFileMenuPage

wpAddFileMenuPage - Syntax

This instance method is called to allow the object to add the *Menu* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND              hwndNotebook;  /*  Settings notebook handle. */
ULONG             rc;            /*  Page identifier. */

rc = _wpAddFileMenuPage(somSelf, hwndNotebook);
```

wpAddFileMenuPage Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpAddFileMenuPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddFileMenuPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFileMenuPage - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFileMenuPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddFileMenuPage - How to Override

This method should always be overridden in order to replace or remove the *Menu* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddFileMenuPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddFileMenuPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddFileMenuPage(MyObj *somSelf,
                  HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
```

```
    MyObjMethodDebug("MyObj","MyObj_wpAddFileMenuPage");

/*  return (parent_wpAddFileMenuPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddFileMenuPage - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddFileTypePage

wpAddFileTypePage - Syntax

This instance method is called to allow the object to add the *Type* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDataFile     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND            hwndNotebook;  /*  Settings notebook handle. */
ULONG           rc;            /*  Page identifier. */

rc = _wpAddFileTypePage(somSelf, hwndNotebook);
```

wpAddFileTypePage Parameter - somSelf

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

wpAddFileTypePage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddFileTypePage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFileTypePage - Parameters

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFileTypePage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddFileTypePage - How to Override

This method should always be overridden in order to replace or remove the *Type* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddFileTypePage - Related Methods

Related Methods

- wpAddSettingsPages

- wpQueryType
- wpSetTitle

wpAddFileTypePage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddFileTypePage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddFileTypePage");

/*  return (parent_wpAddFileTypePage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddFileTypePage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddFirstChild

wpAddFirstChild - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to add one child to a folder.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
WPObject     *Object;   /*  Pointer to the instantiated child object. */

Object = _wpAddFirstChild(somSelf);
```

wpAddFirstChild Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpAddFirstChild Return Value - Object

Object (WPObject *) - returns
Pointer to the instantiated child object.

wpAddFirstChild - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

Object (WPObject *) - returns
Pointer to the instantiated child object.

wpAddFirstChild - Remarks

This method is called when a folder appears in a tree view. As opposed to wpPopulate, wpAddFirstChild adds only one child to the folder; therefore, when the folder appears in the tree view, it has the correct expansion emphasis. When the folder is expanded, it will be fully populated.

wpAddFirstChild - How to Override

This method should be overridden only if there is another way of adding the first child of a folder.

wpAddFirstChild - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override

wpAddFolderBackgroundPage

wpAddFolderBackgroundPage - Syntax

This instance method is called to allow the object to add the *Background* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddFolderBackgroundPage(somSelf, hwndNotebook);
```

wpAddFolderBackgroundPage Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpAddFolderBackgroundPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddFolderBackgroundPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFolderBackgroundPage - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFolderBackgroundPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddFolderBackgroundPage - How to Override

This method should always be overridden in order to replace or remove the *Background* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddFolderBackgroundPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddFolderBackgroundPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddFolderBackgroundPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddFolderBackgroundPage");

/*  return (parent_wpAddFolderBackgroundPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddFolderBackgroundPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddFolderIncludePage

wpAddFolderIncludePage - Syntax

This instance method is called to allow the object to add the *Include* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddFolderIncludePage(somSelf, hwndNotebook);
```

wpAddFolderIncludePage Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpAddFolderIncludePage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddFolderIncludePage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFolderIncludePage - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFolderIncludePage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddFolderIncludePage - How to Override

This method should always be overridden in order to replace or remove the *Include* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddFolderIncludePage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddFolderIncludePage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddFolderIncludePage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddFolderIncludePage");

/*  return (parent_wpAddFolderIncludePage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddFolderIncludePage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddFolderMenu2Page

wpAddFolderMenu2Page - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method adds the standard Menu2 page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddFolderMenu2Page(somSelf, hwndNotebook);
```

wpAddFolderMenu2Page Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpAddFolderMenu2Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddFolderMenu2Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddFolderMenu2Page - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddFolderMenu2Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddFolderMenu2Page - How to Override

This method should always be overridden in order to replace or remove the Menu page 2 from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddFolderMenu2Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddFolderMenu2Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddFolderMenu2Page(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddFolderMenu2Page");

/*  return (parent_wpAddFolderMenu2Page(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddFolderMenu2Page - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddFolderSelfClosePage

wpAddFolderSelfClosePage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method adds the *Folder Automatic Close* page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>
```

```
WPFolder     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Handle of the Settings notebook. */
ULONG         ulPageId;      /*  Identifier for the inserted page. */

ulPageId = _wpAddFolderSelfClosePage(somSelf,
             hwndNotebook);
```

wpAddFolderSelfClosePage Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpAddFolderSelfClosePage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Handle of the Settings notebook.

wpAddFolderSelfClosePage Return Value - ulPageId

ulPageId (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the Folder Automatic Close page was not added to the Settings notebook.

wpAddFolderSelfClosePage - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

hwndNotebook (HWND) - input
Handle of the Settings notebook.

ulPageId (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the Folder Automatic Close page was not added to the Settings notebook.

wpAddFolderSelfClosePage - How to Override

This method is always overridden to replace or remove the *Folder Automatic Close* page from an object that is a descendant of WPFolder. An override of this method does not call the parent.

wpAddFolderSelfClosePage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddFolderSelfClosePage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpAddFolderSortPage

wpAddFolderSortPage - Syntax

This instance method is called to allow the object to add the *Sort* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddFolderSortPage(somSelf, hwndNotebook);
```

wpAddFolderSortPage Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpAddFolderSortPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddFolderSortPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFolderSortPage - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFolderSortPage - Usage

This method must only be called from within an override of the wpAddSettingsPages method.

wpAddFolderSortPage - How to Override

This method should always be overridden in order to replace or remove the *Sort* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddFolderSortPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddFolderSortPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddFolderSortPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddFolderSortPage");

/*  return (parent_wpAddFolderSortPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddFolderSortPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddFolderView1Page

wpAddFolderView1Page - Syntax

This instance method is called to allow the object to add the *View 1* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddFolderView1Page(somSelf, hwndNotebook);
```

wpAddFolderView1Page Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpAddFolderView1Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddFolderView1Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFolderView1Page - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFolderView1Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddFolderView1Page - How to Override

This method should always be overridden in order to replace or remove the *View 1* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddFolderView1Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddFolderView1Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddFolderView1Page(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddFolderView1Page");

/*  return (parent_wpAddFolderView1Page(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddFolderView1Page - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddFolderView2Page

wpAddFolderView2Page - Syntax

This instance method is called to allow the object to add the *View 2* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddFolderView2Page(somSelf, hwndNotebook);
```

wpAddFolderView2Page Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpAddFolderView2Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddFolderView2Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFolderView2Page - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFolderView2Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddFolderView2Page - How to Override

This method should always be overridden in order to replace or remove the *View 2* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddFolderView2Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddFolderView2Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddFolderView2Page(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddFolderView2Page");

/*  return (parent_wpAddFolderView2Page(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddFolderView2Page - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage

Example Code
Related Methods
Glossary

wpAddFolderView3Page

wpAddFolderView3Page - Syntax

This instance method is called to allow the object to add the *View 3* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddFolderView3Page(somSelf, hwndNotebook);
```

wpAddFolderView3Page Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpAddFolderView3Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddFolderView3Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFolderView3Page - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddFolderView3Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddFolderView3Page - How to Override

This method should always be overridden in order to replace or remove the *View 3* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddFolderView3Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddFolderView3Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddFolderView3Page(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddFolderView3Page");

/*  return (parent_wpAddFolderView3Page(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddFolderView3Page - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddKeyboardMappingsPage

wpAddKeyboardMappingsPage - Syntax

This instance method is called to allow the object to add the *Mappings* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPKeyboard     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND            hwndNotebook;  /*  Settings notebook handle. */
ULONG           rc;            /*  Page identifier. */

rc = _wpAddKeyboardMappingsPage(somSelf, hwndNotebook);
```

wpAddKeyboardMappingsPage Parameter - somSelf

somSelf (WPKeyboard *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPKeyboard.

wpAddKeyboardMappingsPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddKeyboardMappingsPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddKeyboardMappingsPage - Parameters

somSelf (WPKeyboard *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPKeyboard.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddKeyboardMappingsPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddKeyboardMappingsPage - How to Override

This method should always be overridden in order to replace or remove the *Mappings* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddKeyboardMappingsPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddKeyboardMappingsPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddKeyboardMappingsPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddKeyboardMappingsPage");

/*  return (parent_wpAddKeyboardMappingsPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddKeyboardMappingsPage - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddKeyboardSpecialNeedsPage

wpAddKeyboardSpecialNeedsPage - Syntax

This instance method is called to allow the object to add the *Special Needs* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPKeyboard     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND            hwndNotebook;  /*  Settings notebook handle. */
ULONG           rc;            /*  Page identifier. */

rc = _wpAddKeyboardSpecialNeedsPage(somSelf,
       hwndNotebook);
```

wpAddKeyboardSpecialNeedsPage Parameter - somSelf

somSelf (WPKeyboard *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPKeyboard.

wpAddKeyboardSpecialNeedsPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddKeyboardSpecialNeedsPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddKeyboardSpecialNeedsPage - Parameters

somSelf (WPKeyboard *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPKeyboard.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddKeyboardSpecialNeedsPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddKeyboardSpecialNeedsPage - How to Override

This method should always be overridden in order to replace or remove the *Special Needs* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddKeyboardSpecialNeedsPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddKeyboardSpecialNeedsPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddKeyboardSpecialNeedsPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddKeyboardSpecialNeedsPage");

/*  return (parent_wpAddKeyboardSpecialNeedsPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddKeyboardSpecialNeedsPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddKeyboardTimingPage

wpAddKeyboardTimingPage - Syntax

This instance method is called to allow the object to add the *Timing* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>
```

```
WPKeyboard     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND            hwndNotebook;  /*  Settings notebook handle. */
ULONG           rc;            /*  Page identifier. */

rc = _wpAddKeyboardTimingPage(somSelf, hwndNotebook);
```

wpAddKeyboardTimingPage Parameter - somSelf

somSelf (WPKeyboard *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPKeyboard.

wpAddKeyboardTimingPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddKeyboardTimingPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddKeyboardTimingPage - Parameters

somSelf (WPKeyboard *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPKeyboard.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddKeyboardTimingPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddKeyboardTimingPage - How to Override

This method should always be overridden in order to replace or remove the *Timing* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddKeyboardTimingPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddKeyboardTimingPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddKeyboardTimingPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddKeyboardTimingPage");

/*  return (parent_wpAddKeyboardTimingPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddKeyboardTimingPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddLaunchPadPage1

wpAddLaunchPadPage1 - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to add the *Toolbar* page 1 to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND             hwndNotebook;  /*  Handle to the Settings notebook. */
ULONG            rc;            /*  Page identifier. */

rc = _wpAddLaunchPadPage1(somSelf, hwndNotebook);
```

wpAddLaunchPadPage1 Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpAddLaunchPadPage1 Parameter - hwndNotebook

hwndNotebook (HWND) - input
Handle to the Settings notebook.

wpAddLaunchPadPage1 Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddLaunchPadPage1 - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

hwndNotebook (HWND) - input
Handle to the Settings notebook.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddLaunchPadPage1 - How to Override

This method should always be overridden to replace or remove the *Toolbar* page 1 from the Settings notebook of an object which is a descendant of WPLaunchPad. An override of this method does not call the parent.

wpAddLaunchPadPage1 - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddLaunchPadPage1 - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpAddLaunchPadPage2

wpAddLaunchPadPage2 - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to add the *Toolbar* page 2 to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
```

```
HWND            hwndNotebook;  /*  Handle to the Settings notebook. */
ULONG           rc;            /*  Page identifier. */

rc = _wpAddLaunchPadPage2(somSelf, hwndNotebook);
```

wpAddLaunchPadPage2 Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpAddLaunchPadPage2 Parameter - hwndNotebook

hwndNotebook (HWND) - input
Handle to the Settings notebook.

wpAddLaunchPadPage2 Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddLaunchPadPage2 - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

hwndNotebook (HWND) - input
Handle to the Settings notebook.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddLaunchPadPage2 - How to Override

This method should always be overridden to replace or remove the *Toolbar* page 2 from the Settings notebook of an object which is a descendant of WPLaunchPad. An override of this method does not call the parent.

wpAddLaunchPadPage2 - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddLaunchPadPage2 - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpAddMouseCometPage

wpAddMouseCometPage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method adds the *Comet Cursor* page to the Setting notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPMouse     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Handle of the Settings notebook. */
ULONG        ulPageId;      /*  Identifier for the inserted page. */

ulPageId = _wpAddMouseCometPage(somSelf, hwndNotebook);
```

wpAddMouseCometPage Parameter - somSelf

somSelf (WPMouse *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPMouse.

wpAddMouseCometPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Handle of the Settings notebook.

wpAddMouseCometPage Return Value - ulPageId

ulPageId (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the *Comet Cursor* page was not added to the Settings notebook.

wpAddMouseCometPage - Parameters

somSelf (WPMouse *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPMouse.

hwndNotebook (HWND) - input
Handle of the Settings notebook.

ulPageId (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the *Comet Cursor* page was not added to the Settings notebook.

wpAddMouseCometPage - Usage

This method is called only during wpAddSettingsPages processing.

wpAddMouseCometPage - How to Override

This method is always overridden to replace or remove the *Comet Cursor* page from an object that is a descendant of WPMouse. An override of this method does not call the parent.

wpAddMouseCometPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddMouseCometPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpAddMouseMappingsPage

wpAddMouseMappingsPage - Syntax

This instance method is called to allow the object to add the *Mappings* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPMouse     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Settings notebook handle. */
ULONG        rc;            /*  Page identifier. */

rc = _wpAddMouseMappingsPage(somSelf, hwndNotebook);
```

wpAddMouseMappingsPage Parameter - somSelf

somSelf (WPMouse *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPMouse.

wpAddMouseMappingsPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddMouseMappingsPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddMouseMappingsPage - Parameters

somSelf (WPMouse *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPMouse.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddMouseMappingsPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddMouseMappingsPage - How to Override

The method should always be overridden in order to replace or remove the *Mappings* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddMouseMappingsPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddMouseMappingsPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddMouseMappingsPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddMouseMappingsPage");

/*  return (parent_wpAddMouseMappingsPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddMouseMappingsPage - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddMousePtrPage

wpAddMousePtrPage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method adds the Mouse *Pointer Customization* page to the Setting notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPMouse     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Handle of the Settings notebook. */
ULONG        ulPageId;      /*  Identifier for the inserted page. */

ulPageId = _wpAddMousePtrPage(somSelf, hwndNotebook);
```

wpAddMousePtrPage Parameter - somSelf

somSelf (WPMouse *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPMouse.

wpAddMousePtrPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Handle of the Settings notebook.

wpAddMousePtrPage Return Value - ulPageId

ulPageId (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the *Mouse Pointer Customization* page was not added to the Settings notebook.

wpAddMousePtrPage - Parameters

somSelf (WPMouse *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPMouse.

hwndNotebook (HWND) - input
Handle of the Settings notebook.

ulPageId (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the *Mouse Pointer Customization* page was not added to the Settings notebook.

wpAddMousePtrPage - Usage

This method is called only during wpAddSettingsPages processing.

wpAddMousePtrPage - How to Override

This method is always overridden to replace or remove the *Mouse Pointer Customization* page from an object that is a descendant of WPMouse. An override of this method does not call the parent.

wpAddMousePtrPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddMousePtrPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpAddMouseTimingPage

wpAddMouseTimingPage - Syntax

This instance method is called to allow the object to add the *Timing* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPMouse     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Settings notebook handle. */
ULONG        rc;            /*  Page identifier. */

rc = _wpAddMouseTimingPage(somSelf, hwndNotebook);
```

wpAddMouseTimingPage Parameter - somSelf

somSelf (WPMouse *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPMouse.

wpAddMouseTimingPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddMouseTimingPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddMouseTimingPage - Parameters

somSelf (WPMouse *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPMouse.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddMouseTimingPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddMouseTimingPage - How to Override

This method should always be overridden in order to replace or remove the *Timing* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddMouseTimingPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddMouseTimingPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddMouseTimingPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddMouseTimingPage");

/*  return (parent_wpAddMouseTimingPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddMouseTimingPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddMouseTypePage

wpAddMouseTypePage - Syntax

This instance method is called to allow the object to add the *Type* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPMouse     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Settings notebook handle. */
ULONG        rc;            /*  Page identifier. */

rc = _wpAddMouseTypePage(somSelf, hwndNotebook);
```

wpAddMouseTypePage Parameter - somSelf

somSelf (WPMouse *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPMouse.

wpAddMouseTypePage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddMouseTypePage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddMouseTypePage - Parameters

somSelf (WPMouse *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPMouse.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddMouseTypePage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddMouseTypePage - How to Override

This method should always be overridden in order to replace or remove the *Type* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddMouseTypePage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddMouseTypePage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddMouseTypePage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddMouseTypePage");

/*  return (parent_wpAddMouseTypePage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddMouseTypePage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddNetworkPage

wpAddNetworkPage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to allow the object to add the *Network* page to the remote printer object's Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPRPrinter     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND            hwndNotebook;  /*  Settings notebook handle. */
BOOL            rc;            /*  Page identifier. */

rc = _wpAddNetworkPage(somSelf, hwndNotebook);
```

wpAddNetworkPage Parameter - somSelf

somSelf (WPRPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPRPrinter.

wpAddNetworkPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddNetworkPage Return Value - rc

rc (BOOL) - returns
Page identifier.

0
Error occurred.
PageId
Identifier for the inserted page.

wpAddNetworkPage - Parameters

somSelf (WPRPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPRPrinter.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (BOOL) - returns
Page identifier.

0
Error occurred.

PageId
Identifier for the inserted page.

wpAddNetworkPage - Usage

This method can be called only from within an override of the wpAddSettingsPages method.

wpAddNetworkPage - How to Override

This method can be overridden to replace or remove the *Network* page from the object's Settings notebook.

wpAddNetworkPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddNetworkPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpAddNetworkPage

wpAddNetworkPage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to add the *Network* details page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPNetgrp     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Handle to the Settings notebook. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddNetworkPage(somSelf, hwndNotebook);
```

wpAddNetworkPage Parameter - somSelf

somSelf (WPNetgrp *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPNetgrp.

wpAddNetworkPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Handle to the Settings notebook.

wpAddNetworkPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0
Error occurred. The *Network* details page was not added to the Settings notebook.

PageId
Identifier for the inserted page.

wpAddNetworkPage - Parameters

somSelf (WPNetgrp *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPNetgrp.

hwndNotebook (HWND) - input
Handle to the Settings notebook.

rc (ULONG) - returns
Page identifier.

0
Error occurred. The *Network* details page was not added to the Settings notebook.

PageId
Identifier for the inserted page.

wpAddNetworkPage - How to Override

This method should always be overridden to replace or remove the *Network* details page from an object which is a descendant of WPNetgrp. An override of this method does not call the parent.

wpAddNetworkPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddNetworkPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpAddObjectGeneralPage

wpAddObjectGeneralPage - Syntax

This instance method is called to allow the object to add the *General* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddObjectGeneralPage(somSelf, hwndNotebook);
```

wpAddObjectGeneralPage Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpAddObjectGeneralPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddObjectGeneralPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddObjectGeneralPage - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddObjectGeneralPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddObjectGeneralPage - How to Override

This method should always be overridden in order to replace or remove the *General* page from the object's Settings notebook. To remove the

page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddObjectGeneralPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddObjectGeneralPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddObjectGeneralPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddObjectGeneralPage");

/*  return (parent_wpAddObjectGeneralPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddObjectGeneralPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddObjectGeneralPage2

wpAddObjectGeneralPage2 - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to allow the object to add the *Animation Icon* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddObjectGeneralPage2(somSelf, hwndNotebook);
```

wpAddObjectGeneralPage2 Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpAddObjectGeneralPage2 Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddObjectGeneralPage2 Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddObjectGeneralPage2 - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddObjectGeneralPage2 - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddObjectGeneralPage2 - How to Override

This method should always be overridden in order to replace or remove the *General* page 2 from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddObjectGeneralPage2 - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddObjectGeneralPage2 - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpAddObjectWindowPage

wpAddObjectWindowPage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method adds the *Standard Options* page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
```

```
HWND          hwndNotebook;  /*  Handle of the Settings notebook. */
ULONG         rc;            /*  Page identifier.       datatype='ULONG'.Page identifier. */

rc = _wpAddObjectWindowPage(somSelf, hwndNotebook);
```

wpAddObjectWindowPage Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpAddObjectWindowPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Handle of the Settings notebook.

wpAddObjectWindowPage Return Value - rc

rc (ULONG) - returns
Page identifier. datatype='ULONG'.Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddObjectWindowPage - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndNotebook (HWND) - input
Handle of the Settings notebook.

rc (ULONG) - returns
Page identifier. datatype='ULONG'.Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddObjectWindowPage - How to Override

This method is always overridden to replace or remove the *Standard Options* page from an object that is a descendant of WPObject. An override of this method does not call the parent.

wpAddObjectWindowPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddObjectWindowPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpAddPowerPage

wpAddPowerPage - Syntax

This method is specific to version 2.1, or higher, of the OS/2 operating system.

This instance method adds the *Standard Power* page to the Settings notebook of the power object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPower     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Settings notebook handle. */
ULONG        rc;            /*  Page identifier. */

rc = _wpAddPowerPage(somSelf, hwndNotebook);
```

wpAddPowerPage Parameter - somSelf

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

wpAddPowerPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddPowerPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddPowerPage - Parameters

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddPowerPage - Related Methods

Related Methods

- wpAddPowerPage
- wpAddPowerViewPage
- wpChangePowerState
- wpQueryAutoRefresh
- wpQueryPowerConfirmation
- wpQueryPowerManagement
- wpQueryRefreshRate
- wpSetAutoRefresh
- wpSetPowerConfirmation
- wpSetPowerManagement
- wpSetRefreshRate

wpAddPowerPage - Example Code

This example overrides this method to eliminate this page from the object's settings.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddPowerPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddPowerPage");

/*  return (parent_wpAddPowerPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddPowerPage - Topics

Select an item:

Syntax
Parameters
Returns
Example Code
Related Methods
Glossary

wpAddPowerViewPage

wpAddPowerViewPage - Syntax

This method is specific to version 2.1, or higher, of the OS/2 operating system.

This instance method adds the *Standard Power View* page to the Settings notebook of the power object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPower     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Settings notebook handle. */
ULONG        rc;            /*  Page identifier. */

rc = _wpAddPowerViewPage(somSelf, hwndNotebook);
```

wpAddPowerViewPage Parameter - somSelf

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

wpAddPowerViewPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddPowerViewPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddPowerViewPage - Parameters

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddPowerViewPage - Related Methods

Related Methods

- wpAddPowerPage
- wpAddPowerViewPage
- wpChangePowerState
- wpQueryAutoRefresh
- wpQueryPowerConfirmation
- wpQueryPowerManagement
- wpQueryRefreshRate

- wpSetAutoRefresh
- wpSetPowerConfirmation
- wpSetPowerManagement
- wpSetRefreshRate

wpAddPowerViewPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddPowerViewPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddPowerViewPage");

/*  return (parent_wpAddPowerViewPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddPowerViewPage - Topics

Select an item:
Syntax
Parameters
Returns
Example Code
Related Methods
Glossary

wpAddProgramAssociationPage

wpAddProgramAssociationPage - Syntax

This instance method is called to allow the object to add the *Association* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgramFile     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND               hwndNotebook;  /*  Settings notebook handle. */
ULONG              rc;            /*  Page identifier. */

rc = _wpAddProgramAssociationPage(somSelf,
       hwndNotebook);
```

wpAddProgramAssociationPage Parameter - somSelf

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

wpAddProgramAssociationPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddProgramAssociationPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddProgramAssociationPage - Parameters

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddProgramAssociationPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddProgramAssociationPage - How to Override

This method should always be overridden in order to replace or remove the *Association* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddProgramAssociationPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddProgramAssociationPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddProgramAssociationPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddProgramAssociationPage");

/*  return (parent_wpAddProgramAssociationPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddProgramAssociationPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddProgramAssociationPage

wpAddProgramAssociationPage - Syntax

This instance method is called to allow the object to add the *Association* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgram     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND           hwndNotebook;  /*  Settings notebook handle. */
ULONG          rc;            /*  Page identifier. */

rc = _wpAddProgramAssociationPage(somSelf,
       hwndNotebook);
```

wpAddProgramAssociationPage Parameter - somSelf

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

wpAddProgramAssociationPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddProgramAssociationPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddProgramAssociationPage - Parameters

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddProgramAssociationPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddProgramAssociationPage - How to Override

This method should always be overridden in order to replace or remove the *Association* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddProgramAssociationPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddProgramAssociationPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddProgramAssociationPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddProgramAssociationPage");

/*  return (parent_wpAddProgramAssociationPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddProgramAssociationPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddProgramPage

wpAddProgramPage - Syntax

This instance method is called to allow the object to add the *Program* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgram     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND           hwndNotebook;  /*  Settings notebook handle. */
ULONG          rc;            /*  Page identifier. */

rc = _wpAddProgramPage(somSelf, hwndNotebook);
```

wpAddProgramPage Parameter - somSelf

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

wpAddProgramPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddProgramPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddProgramPage - Parameters

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddProgramPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddProgramPage - How to Override

This method should always be overridden in order to replace or remove the *Program* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddProgramPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddProgramPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddProgramPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddProgramPage");

/*  return (parent_wpAddProgramPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddProgramPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddProgramPage

wpAddProgramPage - Syntax

This instance method is called to allow the object to add the *Program* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgramFile     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND               hwndNotebook;  /*  Settings notebook handle. */
ULONG              rc;            /*  Page identifier. */

rc = _wpAddProgramPage(somSelf, hwndNotebook);
```

wpAddProgramPage Parameter - somSelf

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

wpAddProgramPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddProgramPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddProgramPage - Parameters

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddProgramPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddProgramPage - How to Override

This method should always be overridden in order to replace or remove the *Program* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddProgramPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddProgramPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddProgramPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddProgramPage");

/*  return (parent_wpAddProgramPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddProgramPage - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddProgramSessionPage

wpAddProgramSessionPage - Syntax

This instance method is called to allow the object to add the *Session* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgram     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND           hwndNotebook;  /*  Settings notebook handle. */
ULONG          rc;            /*  Page identifier. */

rc = _wpAddProgramSessionPage(somSelf, hwndNotebook);
```

wpAddProgramSessionPage Parameter - somSelf

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

wpAddProgramSessionPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddProgramSessionPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddProgramSessionPage - Parameters

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddProgramSessionPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddProgramSessionPage - How to Override

This method should always be overridden in order to replace or remove the *Session* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddProgramSessionPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddProgramSessionPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddProgramSessionPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddProgramSessionPage");

/*  return (parent_wpAddProgramSessionPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddProgramSessionPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddProgramSessionPage

wpAddProgramSessionPage - Syntax

This instance method is called to allow the object to add the *Session* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgramFile     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND               hwndNotebook;  /*  Settings notebook handle. */
ULONG              rc;            /*  Page identifier. */

rc = _wpAddProgramSessionPage(somSelf, hwndNotebook);
```

wpAddProgramSessionPage Parameter - somSelf

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

wpAddProgramSessionPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddProgramSessionPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddProgramSessionPage - Parameters

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddProgramSessionPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddProgramSessionPage - How to Override

This method should always be overridden in order to replace or remove the *Session* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddProgramSessionPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddProgramSessionPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddProgramSessionPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddProgramSessionPage");

/*  return (parent_wpAddProgramSessionPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddProgramSessionPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddReserveDriveLetterPage

wpAddReserveDriveLetterPage - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method adds the reserve drive letter page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDrives     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddReserveDriveLetterPage(somSelf,
       hwndNotebook);
```

wpAddReserveDriveLetterPage Parameter - somSelf

somSelf (WPDrives *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDrives.

wpAddReserveDriveLetterPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddReserveDriveLetterPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddReserveDriveLetterPage - Parameters

somSelf (WPDrives *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDrives.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddReserveDriveLetterPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddReserveDriveLetterPage - How to Override

This method should always be overridden in order to replace or remove the Reserve Drive Letter page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddReserveDriveLetterPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddReserveDriveLetterPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
 SOM_Scope ULONG   SOMLINK MyObj_wpAddReservedDriveLetterPage(MyObj
*somSelf,
                  HWND hwndNotebook)
 {
     /* MyObjData *somThis = MyObjGetData(somSelf); */
     MyObjMethodDebug("MyObj","MyObj_wpAddReservedDriveLetterPage");

 /*  return (parent_wpAddReservedDriveLetterPage(somSelf,hwndNotebook)); */

     return ( SETTINGS_PAGE_REMOVED );
 }
```

wpAddReserveDriveLetterPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override

Usage
Example Code
Related Methods
Glossary

wpAddResourcePage

wpAddResourcePage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to add the *Resource* details page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSharedDir     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND             hwndNotebook;  /*  Handle to the Settings notebook. */
ULONG            rc;            /*  Page identifier. */

rc = _wpAddResourcePage(somSelf, hwndNotebook);
```

wpAddResourcePage Parameter - somSelf

somSelf (WPSharedDir *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSharedDir.

wpAddResourcePage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Handle to the Settings notebook.

wpAddResourcePage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred. The *Resource* details page was not added to the Settings notebook.

PageId	Identifier for the inserted page.

wpAddResourcePage - Parameters

somSelf (WPSharedDir *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSharedDir.

hwndNotebook (HWND) - input
Handle to the Settings notebook.

rc (ULONG) - returns
Page identifier.

0	Error occurred. The *Resource* details page was not added to the Settings notebook.
PageId	Identifier for the inserted page.

wpAddResourcePage - How to Override

This method should always be overridden in order to replace or remove the *Resource* details page from an object which is a descendant of WPSharedDir. An override of this method does not call the parent.

wpAddResourcePage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddResourcePage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpAddServerPage

wpAddServerPage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method adds the *Server* details page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPServer     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Handle to the Settings notebook. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddServerPage(somSelf, hwndNotebook);
```

wpAddServerPage Parameter - somSelf

somSelf (WPServer *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPServer.

wpAddServerPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Handle to the Settings notebook.

wpAddServerPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred. The *Server* details page was not added to the Settings notebook.
PageId	Identifier for the inserted page.

wpAddServerPage - Parameters

somSelf (WPServer *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPServer.

hwndNotebook (HWND) - input
Handle to the Settings notebook.

rc (ULONG) - returns
Page identifier.

0	Error occurred. The *Server* details page was not added to the Settings notebook.
PageId	Identifier for the inserted page.

wpAddServerPage - How to Override

This method should always be overridden in order to replace or remove the *Server* details page from an object which is a descendant of WPServer. An override of this method does not call the parent.

wpAddServerPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddServerPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpAddSessionPage

wpAddSessionPage - Syntax

This method is specfic to Version 3, or higher, of the OS/2 operating system.

This instance method is called to add the *3.0 Session Details* page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPWinConfig     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND             hwndNotebook;  /*  Handle to the Settings notebook. */
ULONG            rc;            /*  Page identifier. */
```

```
rc = _wpAddSessionPage(somSelf, hwndNotebook);
```

wpAddSessionPage Parameter - somSelf

somSelf (WPWinConfig *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPWinConfig.

wpAddSessionPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Handle to the Settings notebook.

wpAddSessionPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred. The *3.0 Session Details* page was not added to the Settings notebook.
PageId	Identifier for the inserted page.

wpAddSessionPage - Parameters

somSelf (WPWinConfig *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPWinConfig.

hwndNotebook (HWND) - input
Handle to the Settings notebook.

rc (ULONG) - returns
Page identifier.

0	Error occurred. The *3.0 Session Details* page was not added to the Settings notebook.
PageId	Identifier for the inserted page.

wpAddSessionPage - How to Override

This method should always be overridden to replace or remove the 3.0 Session Details page from an object which is a descendant of WPWinConfig. An override of this method does not call the parent.

wpAddSessionPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSessionPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpAddSession31Page

wpAddSession31Page - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to add the *3.1 Session Details* page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPWinConfig     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND             hwndNotebook;  /*  Handle to the Settings notebook. */
ULONG            rc;            /*  Page identifier. */

rc = _wpAddSession31Page(somSelf, hwndNotebook);
```

wpAddSession31Page Parameter - somSelf

somSelf (WPWinConfig *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPWinConfig.

wpAddSession31Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Handle to the Settings notebook.

wpAddSession31Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred. The *3.1 Session Details* page was not added to the Settings notebook.
PageId	Identifier for the inserted page.

wpAddSession31Page - Parameters

somSelf (WPWinConfig *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPWinConfig.

hwndNotebook (HWND) - input
Handle to the Settings notebook.

rc (ULONG) - returns
Page identifier.

0	Error occurred. The *3.1 Session Details* page was not added to the Settings notebook.
PageId	Identifier for the inserted page.

wpAddSession31Page - How to Override

This method should always be overridden to replace or remove the *3.1 Session Details* page from an object which is a descendant of WPWinConfig. An override of this method does not call the parent.

wpAddSession31Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSession31Page - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpAddSetPathPage

wpAddSetPathPage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to allow the object to add the *Set-Path* page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSpool     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Settings notebook handle. */
ULONG        rc;            /*  Page identifier. */

rc = _wpAddSetPathPage(somSelf, hwndNotebook);
```

wpAddSetPathPage Parameter - somSelf

somSelf (WPSpool *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSpool.

wpAddSetPathPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddSetPathPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0
Error occurred
PageId
Identifier for the inserted page.

wpAddSetPathPage - Parameters

somSelf (WPSpool *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSpool.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0
Error occurred
PageId
Identifier for the inserted page.

wpAddSetPathPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddSetPathPage - How to Override

This method can be overridden to replace or remove the *Set-Path* page from the object's Settings notebook.

wpAddSetPathPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSetPathPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddSetPathPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddSetPathPage");

/*  return (parent_wpAddSetPathPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddSetPathPage - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddSetPortRedirPage

wpAddSetPortRedirPage - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method adds the Set Port Redirection page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSpool     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Window handle for the Settings notebook. */
ULONG        rc;            /*  Page identifier. */

rc = _wpAddSetPortRedirPage(somSelf, hwndNotebook);
```

wpAddSetPortRedirPage Parameter - somSelf

somSelf (WPSpool *) - input

Pointer to the object on which the method is being invoked.

Points to an object of class WPSpool.

wpAddSetPortRedirPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Window handle for the Settings notebook.

wpAddSetPortRedirPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddSetPortRedirPage - Parameters

somSelf (WPSpool *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSpool.

hwndNotebook (HWND) - input
Window handle for the Settings notebook.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddSetPortRedirPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddSetPortRedirPage - How to Override

This method should always be overridden in order to replace or remove the Port Redirection page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddSetPortRedirPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSetPortRedirPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddSetPortRedirPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddSetPortRedirPage");

/*  return (parent_wpAddSetPortRedirPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddSetPortRedirPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddSetPriorityPage

wpAddSetPriorityPage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to allow the object to add the *Set-Priority* page to the object's Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>
```

```
WPSpool     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Settings notebook handle. */
ULONG        rc;            /*  Page identifier. */

rc = _wpAddSetPriorityPage(somSelf, hwndNotebook);
```

wpAddSetPriorityPage Parameter - somSelf

somSelf (WPSpool *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSpool.

wpAddSetPriorityPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddSetPriorityPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0
Error occurred
PageId
Identifier for the inserted page.

wpAddSetPriorityPage - Parameters

somSelf (WPSpool *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSpool.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0
Error occurred
PageId
Identifier for the inserted page.

wpAddSetPriorityPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddSetPriorityPage - How to Override

This method can be overridden to replace or remove the *Set-Priority* page from the object's Settings notebook.

wpAddSetPriorityPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSetPriorityPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddSetPriorityPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddSetPriorityPage");

/*  return (parent_wpAddSetPriorityPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddSetPriorityPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddSetSyncPage

wpAddSetSyncPage - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method adds the Set Synchronous page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSpool     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Window handle for the Settings notebook. */
ULONG        rc;            /*  Page identifier. */

rc = _wpAddSetSyncPage(somSelf, hwndNotebook);
```

wpAddSetSyncPage Parameter - somSelf

somSelf (WPSpool *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSpool.

wpAddSetSyncPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Window handle for the Settings notebook.

wpAddSetSyncPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddSetSyncPage - Parameters

somSelf (WPSpool *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSpool.

hwndNotebook (HWND) - input
Window handle for the Settings notebook.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddSetSyncPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddSetSyncPage - How to Override

This method should always be overridden in order to replace or remove the Sync page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddSetSyncPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSetSyncPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddSetSyncPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddSetSyncPage");

/*  return (parent_wpAddSetSyncPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddSetSyncPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddSettingsPages

wpAddSettingsPages - Syntax

This instance method is called to allow the object to add all of its *Settings* pages to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
BOOL          rc;            /*  Success indicator. */

rc = _wpAddSettingsPages(somSelf, hwndNotebook);
```

wpAddSettingsPages Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpAddSettingsPages Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddSettingsPages Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpAddSettingsPages - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpAddSettingsPages - Usage

This method is called only by the system.

wpAddSettingsPages - How to Override

This method is always overridden in order to add pages to or remove pages from the Settings notebook. To add a page to the Settings notebook, a call to wpInsertSettingsPage is required. To remove a page from the Settings notebook, the method that adds the page to the Settings notebook must be overridden and SETTINGS_PAGE_REMOVED is returned without calling its parent method.

When adding a page, the parent method should usually be called first. Calling the parent method first will put pages added by the overriding method at the top of the Settings notebook, above the pages added by ancestor classes. Calling the parent last will put pages added by the overriding at the bottom of the Settings notebook, below the pages added by ancestor classes.

wpAddSettingsPages - Related Methods

Related Methods

- wpInsertSettingsPage

wpAddSettingsPages - Example Code

This example adds a Settings notebook page to let the user modify the Last Worker's name.

```
SOM_Scope BOOL SOMLINK myf_wpAddSettingsPages(MYFILE *somSelf, HWND hwndNotebook)

{
    PAGEINFO pageinfo;

    MYFILEData *somThis = MYFILEGetData(somSelf);
    MYFILEMethodDebug("MYFILE","myf_wpAddSettingsPages");

    parent_wpAddSettingsPages(somSelf, hwndNotebook);

    memset((PCH)&pageinfo,0,sizeof(PAGEINFO));
    pageinfo.cb                 = sizeof(PAGEINFO);
    pageinfo.hwndPage           = NULLHANDLE;
    pageinfo.usPageStyleFlags   = BKA_MAJOR;
    pageinfo.usPageInsertFlags  = BKA_FIRST;
    pageinfo.pfnwp              = LastWorkerDlgProc;
    pageinfo.resid              = hmod;
    pageinfo.dlgid              = IDD_LASTWORKER;
    pageinfo.pszName            = "LastWorker";
    pageinfo.pCreateParams      = somSelf;
    pageinfo.idDefaultHelpPanel = ID_HELP_LASTWORKER;
    pageinfo.pszHelpLibraryName = szHelpLibrary;

    return _wpInsertSettingsPage( somSelf, hwndNotebook,&pageinfo );
}
```

wpAddSettingsPages - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddSetViewPage

wpAddSetViewPage - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method adds the Set View page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSpool     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Window handle for the Settings notebook. */
ULONG        rc;            /*  Page identifier. */

rc = _wpAddSetViewPage(somSelf, hwndNotebook);
```

wpAddSetViewPage Parameter - somSelf

somSelf (WPSpool *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSpool.

wpAddSetViewPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Window handle for the Settings notebook.

wpAddSetViewPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddSetViewPage - Parameters

somSelf (WPSpool *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSpool.

hwndNotebook (HWND) - input
Window handle for the Settings notebook.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddSetViewPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddSetViewPage - How to Override

This method should always be overridden in order to replace or remove the Set View page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddSetViewPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSetViewPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddSetViewPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddSetViewPage");

/*  return (parent_wpAddSetViewPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddSetViewPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddSnoop1Page

wpAddSnoop1Page - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method is called to allow an object to add the Detection Level page to the settings notebook.

```
#define INCL_WORKPLACE
#include <os2.h>

WPHwManager     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND             hwndNotebook;  /*  Settings notebook handle. */
ULONG            rc;            /*  Page identifier. */

rc = _wpAddSnoop1Page(somSelf, hwndNotebook);
```

wpAddSnoop1Page Parameter - somSelf

somSelf (WPHwManager *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPHwManager.

wpAddSnoop1Page Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddSnoop1Page Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddSnoop1Page - Parameters

somSelf (WPHwManager *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPHwManager.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns

Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddSnoop1Page - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddSnoop1Page - How to Override

This method should always be overridden to replace or remove the Detection Level page from the object's Settings notebook. To remove the page from the settings notebook, the override method must return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method must call the wpAddSettingsPages method without calling the parent method.

wpAddSnoop1Page - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSnoop1Page - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG SOMLINK MyObj_wpAddSnoop1Page(MyObj *somSelf,
                                              HWND hwndNotebook)

{
  /* MyObjData *somThis = MyObjGetData(somSelf); */
  MyObjMethodDebug("MyObj","MyObj_wpAddSnoop1Page");

  /* return(parent_wpAddSnoop1Page(somSelf,hwndNotebook)); */

  return(SETTINGS_PAGE_REMOVED);
}
```

wpAddSnoop1Page - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code

Related Methods
Glossary

wpAddSoundSchemePage

wpAddSoundSchemePage - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method adds the Sound Scheme page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSound     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Window handle for the Settings notebook. */
ULONG        rc;            /*  Page identifier. */

rc = _wpAddSoundSchemePage(somSelf, hwndNotebook);
```

wpAddSoundSchemePage Parameter - somSelf

somSelf (WPSound *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSound.

wpAddSoundSchemePage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Window handle for the Settings notebook.

wpAddSoundSchemePage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddSoundSchemePage - Parameters

somSelf (WPSound *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSound.

hwndNotebook (HWND) - input
Window handle for the Settings notebook.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddSoundSchemePage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddSoundSchemePage - How to Override

This method should always be overridden in order to replace or remove the Sound Scheme page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddSoundSchemePage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSoundSchemePage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddSoundSchemePage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddSoundSchemePage");

/*  return (parent_wpAddSetViewPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddSoundSchemePage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddSoundWarningBeepPage

wpAddSoundWarningBeepPage - Syntax

This instance method is called to allow the object to add the *Warning Beep* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSound     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND         hwndNotebook;  /*  Settings notebook handle. */
ULONG        rc;            /*  Page identifier. */

rc = _wpAddSoundWarningBeepPage(somSelf, hwndNotebook);
```

wpAddSoundWarningBeepPage Parameter - somSelf

somSelf (WPSound *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSound.

wpAddSoundWarningBeepPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddSoundWarningBeepPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddSoundWarningBeepPage - Parameters

somSelf (WPSound *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSound.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddSoundWarningBeepPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddSoundWarningBeepPage - How to Override

This method should always be overridden in order to replace or remove the *Warning Beep* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddSoundWarningBeepPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSoundWarningBeepPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddSoundWarningBeepPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddSoundWarningBeepPage");

/*  return (parent_wpAddSoundWarningBeepPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddSoundWarningBeepPage - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddSysFdrDefViewPage

wpAddSysFdrDefViewPage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to add the *Default Folder View* page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSystem     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Handle to the Settings notebook. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddSysFdrDefViewPage(somSelf, hwndNotebook);
```

wpAddSysFdrDefViewPage Parameter - somSelf

somSelf (WPSystem *) - input

Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

wpAddSysFdrDefViewPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Handle to the Settings notebook.

wpAddSysFdrDefViewPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred. The *Default Folder View* page was not added to the Settings notebook.
PageId	Identifier for the inserted page.

wpAddSysFdrDefViewPage - Parameters

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

hwndNotebook (HWND) - input
Handle to the Settings notebook.

rc (ULONG) - returns
Page identifier.

0	Error occurred. The *Default Folder View* page was not added to the Settings notebook.
PageId	Identifier for the inserted page.

wpAddSysFdrDefViewPage - Usage

This method can be called at any time.

wpAddSysFdrDefViewPage - How to Override

This method should always be overridden to replace or remove the *Default Folder View* page from an object which is a descendant of WPSystem. An override of this method does not call the parent.

wpAddSysFdrDefViewPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSysFdrDefViewPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpAddSysFdrMenuPage

wpAddSysFdrMenuPage - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method adds a page to the System Settings notebook to enable and disable OpenDoc menu options.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSystem     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Window handle for the Settings notebook. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddSysFdrMenuPage(somSelf, hwndNotebook);
```

wpAddSysFdrMenuPage Parameter - somSelf

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

wpAddSysFdrMenuPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Window handle for the Settings notebook.

wpAddSysFdrMenuPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddSysFdrMenuPage - Parameters

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

hwndNotebook (HWND) - input
Window handle for the Settings notebook.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddSysFdrMenuPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddSysFdrMenuPage - How to Override

This method should always be overridden in order to replace or remove the Folder Menu page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddSysFdrMenuPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSysFdrMenuPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddSysFdrMenuPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddSysFdrMenuPage");

/*  return (parent_wpAddSysFdrMenuPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddSysFdrMenuPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddSysFdrSelfClosePage

wpAddSysFdrSelfClosePage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to add the *Folder Automatic Close* page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSystem     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Handle to the Settings notebook. */
ULONG         ulPageid;      /*  Identifier for the inserted page. */

ulPageid = _wpAddSysFdrSelfClosePage(somSelf,
             hwndNotebook);
```

wpAddSysFdrSelfClosePage Parameter - somSelf

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

wpAddSysFdrSelfClosePage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Handle to the Settings notebook.

wpAddSysFdrSelfClosePage Return Value - ulPageid

ulPageid (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the *Folder Automatic Close* page was not added to the Settings notebook.

wpAddSysFdrSelfClosePage - Parameters

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

hwndNotebook (HWND) - input
Handle to the Settings notebook.

ulPageid (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the *Folder Automatic Close* page was not added to the Settings notebook.

wpAddSysFdrSelfClosePage - Usage

This method can be called at any time.

wpAddSysFdrSelfClosePage - How to Override

This method should always be overridden to replace or remove the *Folder Automatic Close* page from an object which is a descendant of WPSystem. An override of this method does not call the parent.

wpAddSysFdrSelfClosePage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSysFdrSelfClosePage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpAddSystem256ColorPage

wpAddSystem256ColorPage - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method adds the 256 Color page to the System Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSystem     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Window handle for the Settings notebook. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddSystem256ColorPage(somSelf, hwndNotebook);
```

wpAddSystem256ColorPage Parameter - somSelf

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

wpAddSystem256ColorPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Window handle for the Settings notebook.

wpAddSystem256ColorPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddSystem256ColorPage - Parameters

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

hwndNotebook (HWND) - input
Window handle for the Settings notebook.

rc (ULONG) - returns
Page identifier.

0	Error occurred.
PageId	Identifier for the inserted page.

wpAddSystem256ColorPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddSystem256ColorPage - How to Override

This method should always be overridden in order to replace or remove the 256 Color page from the object's Settings notebook. To remove

the page from the Settings notebook, the override method should return SETTINGS_PAGE_REMOVED without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddSystem256ColorPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSystem256ColorPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddSystem256ColorPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddSystem256ColorPage");

/*  return (parent_wpAddSystem256ColorPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddSystem256ColorPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddSystemConfirmationPage

wpAddSystemConfirmationPage - Syntax

This instance method is called to allow the object to add the *Confirmation* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
```

```
#include <os2.h>

WPSystem     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddSystemConfirmationPage(somSelf,
       hwndNotebook);
```

wpAddSystemConfirmationPage Parameter - somSelf

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

wpAddSystemConfirmationPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddSystemConfirmationPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddSystemConfirmationPage - Parameters

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddSystemConfirmationPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddSystemConfirmationPage - How to Override

This method should always be overridden in order to replace or remove the *Confirmation* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddSystemConfirmationPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSystemConfirmationPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddSystemConfirmationPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddSystemConfirmationPage");

/*  return (parent_wpAddSystemConfirmationPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddSystemConfirmationPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddSystemInputPage

wpAddSystemInputPage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to add the *Input* page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSystem     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Handle to the Settings notebook. */
ULONG         ulPageid;      /*  Identifier for the inserted page. */

ulPageid = _wpAddSystemInputPage(somSelf,
             hwndNotebook);
```

wpAddSystemInputPage Parameter - somSelf

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

wpAddSystemInputPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Handle to the Settings notebook.

wpAddSystemInputPage Return Value - ulPageid

ulPageid (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the *Input* details page was not added to the Settings notebook.

wpAddSystemInputPage - Parameters

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

hwndNotebook (HWND) - input
Handle to the Settings notebook.

ulPageid (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the *Input* details page was not added to the Settings notebook.

wpAddSystemInputPage - Usage

This method can be called at any time.

wpAddSystemInputPage - How to Override

This method should always be overridden to replace or remove the *Input* page from an object which is a descendant of WPSystem. An override of this method does not call the parent.

wpAddSystemInputPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSystemInputPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpAddSystemLogoPage

wpAddSystemLogoPage - Syntax

This instance method is called to allow the object to add the *Logo* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSystem     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddSystemLogoPage(somSelf, hwndNotebook);
```

wpAddSystemLogoPage Parameter - somSelf

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

wpAddSystemLogoPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddSystemLogoPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddSystemLogoPage - Parameters

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

hwndNotebook (HWND) - input

Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddSystemLogoPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddSystemLogoPage - How to Override

This method should always be overridden in order to replace or remove the *Logo* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddSystemLogoPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSystemLogoPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddSystemLogoPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddSystemLogoPage");

/*  return (parent_wpAddSystemLogoPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddSystemLogoPage - Topics

Select an item:
Syntax
Parameters
Returns

How to Override
Usage
Example Code
Related Methods
Glossary

wpAddSystemPrintScreenPage

wpAddSystemPrintScreenPage - Syntax

This instance method is called to allow the object to add the *Print Screen* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSystem     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddSystemPrintScreenPage(somSelf,
       hwndNotebook);
```

wpAddSystemPrintScreenPage Parameter - somSelf

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

wpAddSystemPrintScreenPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddSystemPrintScreenPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred

PageId	Identifier for the inserted page.

wpAddSystemPrintScreenPage - Parameters

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddSystemPrintScreenPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddSystemPrintScreenPage - How to Override

This method should always be overridden in order to replace or remove the *Print Screen* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddSystemPrintScreenPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSystemPrintScreenPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddSystemPrintScreenPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddSystemPrintScreenPage");
```

```
/*  return (parent_wpAddSystemPrintScreenPage(somSelf,hwndNotebook)); */

    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddSystemPrintScreenPage - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddSystemScreenPage

wpAddSystemScreenPage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to add the *Screen* page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSystem     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Window handle to the Settings notebook. */
ULONG         ulPageId;      /*  Identifier for the inserted page. */

ulPageId = _wpAddSystemScreenPage(somSelf,
             hwndNotebook);
```

wpAddSystemScreenPage Parameter - somSelf

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

wpAddSystemScreenPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Window handle to the Settings notebook.

wpAddSystemScreenPage Return Value - ulPageId

ulPageId (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the *Screen* page was not added to the Settings notebook.

wpAddSystemScreenPage - Parameters

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

hwndNotebook (HWND) - input
Window handle to the Settings notebook.

ulPageId (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the *Screen* page was not added to the Settings notebook.

wpAddSystemScreenPage - How to Override

This method should always be overridden to replace or remove the *Screen* page from an object which is a descendant of WPSystem. An override of this method does not call the parent.

wpAddSystemScreenPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSystemScreenPage - Topics

Select an item:
Syntax

Parameters
Returns
How to Override
Related Methods
Glossary

wpAddSystemWindowPage

wpAddSystemWindowPage - Syntax

This instance method is called to allow the object to add the *Window* page to its Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSystem     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Settings notebook handle. */
ULONG         rc;            /*  Page identifier. */

rc = _wpAddSystemWindowPage(somSelf, hwndNotebook);
```

wpAddSystemWindowPage Parameter - somSelf

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

wpAddSystemWindowPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Settings notebook handle.

wpAddSystemWindowPage Return Value - rc

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddSystemWindowPage - Parameters

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

hwndNotebook (HWND) - input
Settings notebook handle.

rc (ULONG) - returns
Page identifier.

0	Error occurred
PageId	Identifier for the inserted page.

wpAddSystemWindowPage - Usage

This method must be called only from within an override of the wpAddSettingsPages method.

wpAddSystemWindowPage - How to Override

This method should always be overridden in order to replace or remove the *Window* page from the object's Settings notebook. To remove the page from the Settings notebook, the override method should return *SETTINGS_PAGE_REMOVED* without calling the parent method. To replace the page with another page, the override method should call the wpInsertSettingsPage method without calling the parent method.

wpAddSystemWindowPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddSystemWindowPage - Example Code

This example overrides the method to eliminate this page from the object's Settings notebook.

```
SOM_Scope ULONG   SOMLINK MyObj_wpAddSystemWindowPage(MyObj *somSelf,
                HWND hwndNotebook)
{
    /* MyObjData *somThis = MyObjGetData(somSelf); */
    MyObjMethodDebug("MyObj","MyObj_wpAddSystemWindowPage");

/*  return (parent_wpAddSystemWindowPage(somSelf,hwndNotebook)); */
```

```
    return ( SETTINGS_PAGE_REMOVED );
}
```

wpAddSystemWindowPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAddTitleConfirmationPage

wpAddTitleConfirmationPage - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to add the *Title Confirmation* page to the Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPSystem     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Window handle for the Settings notebook. */
ULONG         ulPageId;      /*  Identifier for the inserted page. */

ulPageId = _wpAddTitleConfirmationPage(somSelf,
             hwndNotebook);
```

wpAddTitleConfirmationPage Parameter - somSelf

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

wpAddTitleConfirmationPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Window handle for the Settings notebook.

wpAddTitleConfirmationPage Return Value - ulPageId

ulPageId (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the *Title Confirmation* page was not added to the Settings notebook.

wpAddTitleConfirmationPage - Parameters

somSelf (WPSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPSystem.

hwndNotebook (HWND) - input
Window handle for the Settings notebook.

ulPageId (ULONG) - returns
Identifier for the inserted page.

A return value of 0 indicates that the *Title Confirmation* page was not added to the Settings notebook.

wpAddTitleConfirmationPage - How to Override

This method should always be overridden to replace or remove the *Title Confirmation* page from an object which is a descendant of WPSystem. An override of this method does not call the parent.

wpAddTitleConfirmationPage - Related Methods

Related Methods

- wpAddSettingsPages
- wpInsertSettingsPage

wpAddTitleConfirmationPage - Topics

Select an item:
Syntax
Parameters

Returns
How to Override
Related Methods
Glossary

wpAddToContent

wpAddToContent - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method adds an object to a folder's Content List.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
WPObject     *Object;   /*  Pointer to the object to be added to the folder's Content List. */
BOOL          rc;       /*  Success indicator. */

rc = _wpAddToContent(somSelf, Object);
```

wpAddToContent Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpAddToContent Parameter - Object

Object (WPObject *) - input
Pointer to the object to be added to the folder's Content List.

wpAddToContent Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.

FALSE	Error occurred.

wpAddToContent - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

Object (WPObject *) - input
Pointer to the object to be added to the folder's Content List.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpAddToContent - How to Override

This method should be overridden to intercept the notification of the "Added an object to a folder" event for subclasses that define their own folder view. The parent must always be called.

wpAddToContent - Related Methods

Related Methods

- wpDeleteFromContent

wpAddToContent - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpAddToObjUseList

wpAddToObjUseList - Syntax

This instance method is called to add an item type to an object's in-use list.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
PUSEITEM      pUseItem;  /*  Pointer to a USEITEM structure. */
BOOL          rc;        /*  Success indicator. */

rc = _wpAddToObjUseList(somSelf, pUseItem);
```

wpAddToObjUseList Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpAddToObjUseList Parameter - pUseItem

pUseItem (PUSEITEM) - input
Pointer to a USEITEM structure.

wpAddToObjUseList Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpAddToObjUseList - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pUseItem (PUSEITEM) - input
Pointer to a USEITEM structure.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpAddToObjUseList - Remarks

This method will add a specified item type to an object's in-use (USEITEM) list. Every workplace object in the system has an in-use list. The in-use list is a linked list of USEITEM structures which provides the object with important information such as the number of container *(WC_CONTAINER)* windows it has been inserted into. It also provides the number of open views (contents, help, and settings) of itself that already exist and how much memory it has allocated. The USEITEM structure consists of an item type and a pointer to the next USEITEM structure and is followed immediately by an item type-specific structure.

wpAddToObjUseList - Usage

The following types of items can be added to the use list:

USAGE_MEMORY	This item specifies a block of memory allocated for this object through the use of the wpAllocMem method. Items of this type add a MEMORYITEM structure to the end of the USEITEM structure.
USAGE_OPENVIEW	When a view of an object is opened, one of these items is added to the in-use list. If multiple, concurrent views are not enabled for this object, the *USAGE_OPENVIEW* items are used by the system to automatically switch to the open view when the user chooses to open the same view again. This behavior is controlled by the application by calling the wpOpen or the wpSwitchTo method. This item is also used by the system to update the title bar text and switch-entry text when the user changes the object title. If this type is specified, the in-use emphasis bit is turned on for all inserted records for this object. Items of this type concatenate a VIEWITEM structure to the end of the USEITEM structure.
USAGE_RECORD	For every view (*WC_CONTAINER*) window that the object is inserted into, there is one of these items on its in-use list. This enables an object to refresh its appearance in all views at the same time. Items of this type add a RECORDITEM structure to the end of the USEITEM structure.

wpAddToObjUseList - Related Methods

Related Methods

- wpAllocMem
- wpClose
- wpDeleteFromObjUseList
- wpFindUseItem

wpAddToObjUseList - Topics

Select an item:
Syntax
Parameters
Returns

Remarks
Usage
Related Methods
Glossary

wpAddUserItemsToPopupMenu

wpAddUserItemsToPopupMenu - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method adds user defined items to the pop-up menu.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
HWND              hwndMenu;   /*  Handle to the menu. */
HWND              hwndCnr;    /*  Handle to the container. */
ULONG             iPosition;  /*  The position on the pop-up menu. */
BOOL              rc;         /*  Success indicator */

rc = _wpAddUserItemsToPopupMenu(somSelf, hwndMenu,
       hwndCnr, iPosition);
```

wpAddUserItemsToPopupMenu Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpAddUserItemsToPopupMenu Parameter - hwndMenu

hwndMenu (HWND) - input
Handle to the menu.

wpAddUserItemsToPopupMenu Parameter - hwndCnr

hwndCnr (HWND) - input

Handle to the container.

wpAddUserItemsToPopupMenu Parameter - iPosition

iPosition (ULONG) - input
The position on the pop-up menu.

wpAddUserItemsToPopupMenu Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpAddUserItemsToPopupMenu - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

hwndMenu (HWND) - input
Handle to the menu.

hwndCnr (HWND) - input
Handle to the container.

iPosition (ULONG) - input
The position on the pop-up menu.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpAddUserItemsToPopupMenu - Remarks

This method may be made private in a future release. Do not attempt to override or call this method.

wpAddUserItemsToPopupMenu - Usage

This method is called by the Workplace Shell during processing of the wpModifyMenu method. It was not intended for a user to either override

or call this method.

wpAddUserItemsToPopupMenu - How to Override

This method is generally not overridden.

wpAddUserItemsToPopupMenu - Related Methods

Related Methods

- wpFilterMenu
- wpFilterPopupMenu
- wpInsertPopupMenuItems
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyMenu
- wpModifyPopupMenu

wpAddUserItemsToPopupMenu - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpAllocMem

wpAllocMem - Syntax

This instance method is called to allocate memory for use by an object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
ULONG         cbBytes;   /*  Specifies the size, in bytes, of memory required. */
PULONG        prc;       /*  Pointer to error code. */
PBYTE         pbNewMem;  /*  Success indicator. */

pbNewMem = _wpAllocMem(somSelf, cbBytes, prc);
```

wpAllocMem Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpAllocMem Parameter - cbBytes

cbBytes (ULONG) - in/out
Specifies the size, in bytes, of memory required.

wpAllocMem Parameter - prc

prc (PULONG) - in/out
Pointer to error code.

NULL	If an error occurs, the system displays an appropriate message.
prc	If an error occurs, one of the following is returned in *prc*:

6	ERROR_INVALID_HANDLE
8	ERROR_NOT_ENOUGH_MEMORY
95	ERROR_INTERRUPT
103	ERROR_TOO_MANY_SEM_REQUESTS
105	ERROR_SEM_OWNER_DIED
640	ERROR_TIMEOUT

wpAllocMem Return Value - pbNewMem

pbNewMem (PBYTE) - returns
Success indicator.

NULL	Memory allocation failed.
Other	A pointer to the newly allocated memory.

wpAllocMem - Parameters

somSelf (WPObject *) - input

Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

cbBytes (ULONG) - in/out
Specifies the size, in bytes, of memory required.

prc (PULONG) - in/out
Pointer to error code.

NULL	If an error occurs, the system displays an appropriate message.
prc	If an error occurs, one of the following is returned in *prc*:

6	ERROR_INVALID_HANDLE
8	ERROR_NOT_ENOUGH_MEMORY
95	ERROR_INTERRUPT
103	ERROR_TOO_MANY_SEM_REQUESTS
105	ERROR_SEM_OWNER_DIED
640	ERROR_TIMEOUT

pbNewMem (PBYTE) - returns
Success indicator.

NULL	Memory allocation failed.
Other	A pointer to the newly allocated memory.

wpAllocMem - Remarks

Memory allocated by wpAllocMem should be de-allocated when it is no longer needed by calling wpFreeMem. Allocated memory not cleaned up by an object is de-allocated automatically when the object is no longer in use.

This method does not zerofill the allocated memory.

wpAllocMem - Usage

This method can be called at any time to allocate memory to be used for this object only.

wpAllocMem - How to Override

This method should be overridden if a substitute memory allocation facility is to be used. Object classes overriding this method should also override wpFreeMem.

wpAllocMem - Related Methods

Related Methods

- wpFreeMem

wpAllocMem - Example Code

This example initializes state variables and allocates any extra memory needed.

```
SOM_Scope void SOMLINK myf_wpInitData(MYFILE *somSelf)
{
    MYFILEData *somThis = MYFILEGetData(somSelf);
    MYFILEMethodDebug("MYFILE","myf_wpInitData");

    parent_wpInitData(somSelf);    /* Have parent method do initialization */

    _pszLastWorker = (PSZ) _wpAllocMem(somSelf,
                                       CCHMAXPATH+1,  /* Number of bytes */
                                       NULL);         /* Return error message */

    if (_pszLastWorker) {

       strcpy(_pszLastWorker, DEFAULT_LASTWORKER);

    } /* endif */

}
```

wpAllocMem - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpAppendObject

wpAppendObject - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method append a file system object to another file system object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
WPObject     *targetObject;  /*  Object to append to. */
BOOL32        fMove;         /*  Type of move flag. */
BOOL32        rc;            /*  Success indicator */

rc = _wpAppendObject(somSelf, targetObject,
       fMove);
```

wpAppendObject Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpAppendObject Parameter - targetObject

targetObject (WPObject *) - input
Object to append to.

wpAppendObject Parameter - fMove

fMove (BOOL32) - input
Type of move flag.

TRUE	Moves and appends an object.
FALSE	Copies and appends an object.

wpAppendObject Return Value - rc

rc (BOOL32) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpAppendObject - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

targetObject (WPObject *) - input
Object to append to.

fMove (BOOL32) - input
Type of move flag.

TRUE	Moves and appends an object.
FALSE	Copies and appends an object.

rc (BOOL32) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpAppendObject - Usage

This method can be called at any time to append a file to the end of another file.

wpAppendObject - How to Override

This method can be overridden by an object class that wants to provide the capability of combining objects.

wpAppendObject - Related Methods

Related Methods

- wpCopyObject
- wpMoveObject
- wpReplaceObject

wpAppendObject - Example Code

This example appends the file "c:\file2" to the end of "c:\file1" and preserves the original "c:\file2" file.

```
WPObject     *File1,File2;

File1 = _wpQueryObjectFromPath(_WPFileSystem,"c:\\file1");
File2 = _wpQueryObjectFromPath(_WPFileSystem,"c:\\file2");
_wpAppendObject(File2,File1,FALSE);
```

wpAppendObject - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpAssertObjectMutexSem

wpAssertObjectMutexSem - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method verifies that an object's mutex semaphore is held for the current thread.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL          rc;       /*  Success indicator */

rc = _wpAssertObjectMutexSem(somSelf);
```

wpAssertObjectMutexSem Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpAssertObjectMutexSem Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	The object semaphore is held by the calling thread.
FALSE	The object semaphore is not held by the calling thread.

wpAssertObjectMutexSem - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

rc (BOOL) - returns

Success indicator

TRUE	The object semaphore is held by the calling thread.
FALSE	The object semaphore is not held by the calling thread.

wpAssertObjectMutexSem - Remarks

Each object has associated with it a mutex semaphore that can be used to serialize access to resources. This method is called to assert that the current thread indeed holds this mutex semaphore. Methods that expect to be invoked under semaphore protection can use this method to validate that the caller has indeed obtained ownership of the mutex semaphore.

This semaphore is used primarily to control access to the linked list of use items, but it can be used for other purposes, if needed. Folders also have a separate mutex semaphore that is used to protect the content chain for that folder.

If you need both the object and the folder mutex semaphores, then you must obtain the folder mutex first.

If you need multiple object mutex semaphores, obtain them in object address order.

As with most semaphores, hold the semaphore for the minimum possible time.

wpAssertObjectMutexSem - Usage

This method is called to test whether some other method has requested serialization of access to this object. For example, you might define a method that requires the caller to obtain the object's mutex semaphore before calling your method. You can use the wpAssertObjectMutexSem method to insure that the caller has properly serialized the object before calling your method.

wpAssertObjectMutexSem - How to Override

This method is generally not overridden.

wpAssertObjectMutexSem - Related Methods

Related Methods

- wpRequestObjectMutexSem
- wpReleaseObjectMutexSem

wpAssertObjectMutexSem - Example Code

```
BOOL     bIsLocked;

bIsLocked = _wpAssertObjectMutexSem(self);
if (!bIsLocked)
{
   return FALSE;
}
```

wpAssertObjectMutexSem - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpChangePowerState

wpChangePowerState - Syntax

This method is specific to version 2.1, or higher, of the OS/2 operating system.

This instance method performs the power management state change requests to the OS/2 Power Management Subsystem.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPower     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG        ulParm1;  /*  Parameter 1. */
ULONG        ulparm2;  /*  Parameter 2. */
BOOL         rc;       /*  Success indicator. */

rc = _wpChangePowerState(somSelf, ulParm1,
       ulparm2);
```

wpChangePowerState Parameter - somSelf

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

wpChangePowerState Parameter - ulParm1

ulParm1 (ULONG) - input
Parameter 1.

This parameter contains a complex value. The ULONG value is composed of two separate USHORT values. These values are as follows:

SubId (*USHORT*)
The requested change power state identifier. *SubId* defines the type of power state transition being requested.

0x0006	Set Power State Request The set power state request is used to initiate a change in the system's power conservation support state. The *ulparm2* value determines the scope of the request.

Note: All other *SubId* values are reserved for future use.

0 (*USHORT*)
Reserved value, must be 0.

Example of *ulParm1* construction:

```
SubId(USHORT) = 0006h
ulParm1 = MAKEULONG(SubId,0)
```

wpChangePowerState Parameter - ulparm2

ulparm2 (ULONG) - input
Parameter 2.

This parameter contains a complex value. The ULONG value is composed of two separate USHORT values.

DevId (*USHORT*)
Requested device identifier.

0x0001	All system-managed services. System-managed devices are the hardware-system specific set of devices that support power-managed states. Because this set of devices is hardware specific, each system configuration may have unique support capabilities.

Note: All other *DevId* values are reserved for future use.

PwrState (*USHORT*)
Requested power state that the devices specified in *DevId* are to be set to.

0x0001	Standby State Request. This state indicates that the specified devices should be set into a power conservation mode that consists of a partial power level that allows for improved energy conservation without noticeable power up delays for re-established operation.
0x0002	Suspend State Request. This state indicates that the specified devices should be set into a power conservation mode that consists of a partial power level that allows for substantial energy conservation with accepted noticeable power-up delays for re-established operation.

Note: All other *PwrState* values are reserved for future use.

Example of *ulparm2* construction:

```
DevId(USHORT) = 0001h
PwrState(USHORT) = 0002h
ulParm2 = MAKEULONG(DevId,PwrState)
```

wpChangePowerState Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	The request was successful.
FALSE	The request was unsuccessful.

wpChangePowerState - Parameters

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

ulParm1 (ULONG) - input
Parameter 1.

This parameter contains a complex value. The ULONG value is composed of two separate USHORT values. These values are as follows:

SubId (*USHORT*)
The requested change power state identifier. *SubId* defines the type of power state transition being requested.

0x0006	Set Power State Request The set power state request is used to initiate a change in the system's power conservation support state. The *ulparm2* value determines the scope of the request.

Note: All other *SubId* values are reserved for future use.

0 (*USHORT*)
Reserved value, must be 0.

Example of *ulParm1* construction:

```
SubId(USHORT) = 0006h
ulParm1 = MAKEULONG(SubId,0)
```

ulparm2 (ULONG) - input
Parameter 2.

This parameter contains a complex value. The ULONG value is composed of two separate USHORT values.

DevId (*USHORT*)
Requested device identifier.

0x0001	All system-managed services. System-managed devices are the hardware-system specific set of devices that

support power-managed states. Because this set of devices is hardware specific, each system configuration may have unique support capabilities.

Note: All other *DevId* values are reserved for future use.

PwrState (*USHORT*)
Requested power state that the devices specified in *DevId* are to be set to.

0x0001	Standby State Request. This state indicates that the specified devices should be set into a power conservation mode that consists of a partial power level that allows for improved energy conservation without noticeable power up delays for re-established operation.
0x0002	Suspend State Request. This state indicates that the specified devices should be set into a power conservation mode that consists of a partial power level that allows for substantial energy conservation with accepted noticeable power-up delays for re-established operation.

Note: All other *PwrState* values are reserved for future use.

Example of *ulparm2* construction:

```
DevId(USHORT) = 0001h
PwrState(USHORT) = 0002h
ulParm2 = MAKEULONG(DevId,PwrState)
```

rc (BOOL) - returns
Success indicator.

TRUE	The request was successful.
FALSE	The request was unsuccessful.

wpChangePowerState - Related Methods

Related Methods

- wpAddPowerPage
- wpAddPowerViewPage
- wpChangePowerState
- wpQueryAutoRefresh
- wpQueryPowerConfirmation
- wpQueryPowerManagement
- wpQueryRefreshRate
- wpSetAutoRefresh
- wpSetPowerConfirmation
- wpSetPowerManagement
- wpSetRefreshRate

wpChangePowerState - Topics

Select an item:
Syntax
Parameters
Returns
Related Methods
Glossary

wpClose

wpClose - Syntax

This instance method is called to close all open views of an object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL          rc;       /*  Success indicator. */

rc = _wpClose(somSelf);
```

wpClose Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpClose Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpClose - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpClose - Remarks

This method will cycle through the object's in-use list that was created by calls to the wpAddToObjUseList method. All open windows specified by *USAGE_OPENVIEW* items send a WM_CLOSE message. Running executables specified by *USAGE_OPENVIEW* items are terminated.

wpClose - Usage

This method should be called to close all open views of the object and to free all allocated resources.

wpClose - How to Override

If this method is overridden, it should call the parent last to ensure that allocated resources are properly deallocated.

wpClose - Related Methods

Related Methods

- wpAddToObjUseList
- wpDeleteFromObjUseList
- wpFindUseItem
- wpOpen

wpClose - Example Code

In this example, when a folder object is closed, we will zero out its first view instance variable so that the next time it has a first view opened, we can keep track of that.

```
SOM_Scope BOOL   SOMLINK myfold_wpClose(MYFOLDER *somSelf)

{
    MYFOLDERData *somThis = MYFOLDERGetData(somSelf);
    BOOL          rcParentClose;
    MYFOLDERMethodDebug("MYFOLDER","myfold_wpClose");

    rcParentClose = parent_wpClose(somSelf);  /* Call parent method first */

    _setFirstViewHandle(somSelf, NULLHANDLE);  /* Do our thing */

    return( rcParentClose );
}
```

wpClose - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpCnrDeleteUseItem

wpCnrDeleteUseItem - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to delete the object's USAGE_RECORD use-item for the container specified; it does not remove the record from the container. To delete the USAGE_RECORD use-item and remove the record from the container, use wpCnrRemoveObject.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
HWND          hwndCnr;  /*  Handle to the container. */
BOOL          rc;       /*  Success indicator. */

rc = _wpCnrDeleteUseItem(somSelf, hwndCnr);
```

wpCnrDeleteUseItem Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpCnrDeleteUseItem Parameter - hwndCnr

hwndCnr (HWND) - input
Handle to the container.

wpCnrDeleteUseItem Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpCnrDeleteUseItem - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndCnr (HWND) - input
Handle to the container.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpCnrDeleteUseItem - Related Methods

Related Methods

- wpAddToObjUseList
- wpDeleteFromObjUseList
- wpFindUseItem
- wpclsInsertMultipleObjects
- wpclsRemoveObjects

wpCnrDeleteUseItem - Topics

Select an item:
Syntax
Parameters
Returns
Related Methods
Glossary

wpCnrInsertObject

wpCnrInsertObject - Syntax

This instance method inserts a record into a container control window.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject          *somSelf;      /*  Pointer to the object on which the method is being invoked. */
HWND               hwndCnr;      /*  Handle of container control window. */
PPOINTL            pptlIcon;     /*  Initial icon position in the container control window. */
PMINIRECORDCORE    preccParent;  /*  Pointer to the parent record. */
PRECORDINSERT      pRecInsert;   /*  Record position. */
PMINIRECORDCORE    pRecord;      /*  Pointer to the inserted record. */

pRecord = _wpCnrInsertObject(somSelf, hwndCnr,
            pptlIcon, preccParent, pRecInsert);
```

wpCnrInsertObject Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpCnrInsertObject Parameter - hwndCnr

hwndCnr (HWND) - input
Handle of container control window.

Possible values are described in the following list:

NULLHANDLE
A pop-up request is made on the "whitespace" of an open view.
other
The handle of the container requesting the pop-up menu.

wpCnrInsertObject Parameter - pptlIcon

pptlIcon (PPOINTL) - input
Initial icon position in the container control window.

wpCnrInsertObject Parameter - preccParent

preccParent (PMINIRECORDCORE) - input
Pointer to the parent record.

preccParent specifies the record of the immediate parent of the record being inserted. This parameter should be set to NULL if the record has no parent or if tree view is not supported.

wpCnrInsertObject Parameter - pRecInsert

pRecInsert (PRECORDINSERT) - input
Record position.

This parameter specifies how this record is to be inserted relative to other records in the same container.

NULL	Insert a record into the next available position.
Other	Insert a record into the position specified by *pRecInsert*.

wpCnrInsertObject Return Value - pRecord

pRecord (PMINIRECORDCORE) - returns
Pointer to the inserted record.

A return value of NULL indicates that an error occurred.

wpCnrInsertObject - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndCnr (HWND) - input
Handle of container control window.

Possible values are described in the following list:

NULLHANDLE
A pop-up request is made on the "whitespace" of an open view.
other
The handle of the container requesting the pop-up menu.

pptlIcon (PPOINTL) - input
Initial icon position in the container control window.

preccParent (PMINIRECORDCORE) - input
Pointer to the parent record.

preccParent specifies the record of the immediate parent of the record being inserted. This parameter should be set to NULL if the record has no parent or if tree view is not supported.

pRecInsert (PRECORDINSERT) - input
Record position.

This parameter specifies how this record is to be inserted relative to other records in the same container.

NULL	Insert a record into the next available position.
Other	Insert a record into the position specified by *pRecInsert*.

pRecord (PMINIRECORDCORE) - returns
Pointer to the inserted record.

A return value of NULL indicates that an error occurred.

wpCnrInsertObject - Remarks

This method inserts an object into a container control window (WC_CONTAINER). These container windows can be created by an application or by the system (such as those in folders and Settings notebooks). Objects inserted using this method automatically inherit the default Workplace Shell behavior, such as context menus, and drag and drop capabilities.

When an object is inserted into a container, the Workplace Shell adds a USAGE_RECORD item to the object's in-use list. There will be one USAGE_RECORD item for each view (container control window) that the object is inserted into. See wpAddToObjUseList for more information on an object's in-use list.

This method sends the CM_INSERTRECORD message to the container control window.

wpCnrRemoveObject should be called to remove each record before the container window is destroyed.

The OBJECT_FROM_PREC (prec) macro can be used to obtain a pointer to the object (WPObject *) associated with a given MINIRECORDCORE structure that was inserted using wpCnrInsertObject.

The USER_FROM_PREC(prec) macro can be used to access the application-definable 32-bit field within the MINIRECORDCORE structure created when an object is put into a list control using wpCnrInsertObject.

wpCnrInsertObject - Usage

This method is used to give Workplace Shell object behavior (such as context menu support) to records inserted directly into a *WC_CONTAINER* container control window. To remove the record from the container, a call should be made to wpCnrRemoveObject.

wpCnrInsertObject - How to Override

This method is generally not overridden.

wpCnrInsertObject - Related Methods

Related Methods

- wpAddToObjUseList
- wpCnrRemoveObject
- wpclsInsertMultipleObjects

wpCnrInsertObject - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpCnrRefreshDetails

wpCnrRefreshDetails - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method causes all currently visible RECORDCORE structures to be refreshed with the current object details.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */

wpCnrRefreshDetails(somSelf);
```

wpCnrRefreshDetails Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpCnrRefreshDetails - Return Value

There is no return value for this method.

wpCnrRefreshDetails - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

There is no return value for this method.

wpCnrRefreshDetails - Usage

This method can be called at any time to update current object details.

wpCnrRefreshDetails - How to Override

This method is generally not overridden.

wpCnrRefreshDetails - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Glossary

wpCnrRemoveObject

wpCnrRemoveObject - Syntax

This instance method is called to remove a record from a container control window.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
HWND          hwndCnr;  /*  Handle of the container control (WC_CONTAINER) window. */
BOOL          rc;       /*  Success indicator. */
```

```
rc = _wpCnrRemoveObject(somSelf, hwndCnr);
```

wpCnrRemoveObject Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpCnrRemoveObject Parameter - hwndCnr

hwndCnr (HWND) - input
Handle of the container control (*WC_CONTAINER*) window.

wpCnrRemoveObject Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpCnrRemoveObject - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndCnr (HWND) - input
Handle of the container control (*WC_CONTAINER*) window.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpCnrRemoveObject - Remarks

When an object is removed from a container, the USAGE_RECORD item that was added to the in-use list when the object was inserted is

removed.

wpCnrRemoveObject - Usage

This method removes records inserted by calling either wpCnrInsertObject or wpclsInsertMultipleObjects.

wpCnrRemoveObject - How to Override

This method is generally not overridden.

wpCnrRemoveObject - Related Methods

Related Methods

- wpCnrInsertObject
- wpDeleteFromObjUseList
- wpclsInsertMultipleObjects

wpCnrRemoveObject - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpCnrSetEmphasis

wpCnrSetEmphasis - Syntax

This instance method is called to allow the object to change its visual emphasis.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
```

```
ULONG         ulEmphasisAttr;  /*  Record attributes of the container. */
BOOL          fTurnOn;         /*  Flag indicating whether the specified attributes are to be set or reset. */
BOOL          rc;              /*  Success indicator. */

rc = _wpCnrSetEmphasis(somSelf, ulEmphasisAttr,
       fTurnOn);
```

wpCnrSetEmphasis Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpCnrSetEmphasis Parameter - ulEmphasisAttr

ulEmphasisAttr (ULONG) - input
Record attributes of the container.

Any or all of the following flags can be combined using the logical OR (|) operator:

CRA_CURSORED
Specifies that a record will be drawn with a selection cursor.

CRA_DISABLED
Specifies that a record will be drawn with unavailable-state emphasis.

CRA_INUSE
Specifies that a record will be drawn with in-use emphasis.

CRA_PICKED
Specifies that the record will be drawn with "picked" emphasis; for example, it is part of a lazy drag set.

CRA_SELECTED
Specifies that a record will be drawn with selected-state emphasis.

CRA_SOURCE
Specifies that a record will be drawn with source-menu emphasis.

wpCnrSetEmphasis Parameter - fTurnOn

fTurnOn (BOOL) - input
Flag indicating whether the specified attributes are to be set or reset.

TRUE	Set the specified attributes.
FALSE	Reset the specified attributes.

wpCnrSetEmphasis Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpCnrSetEmphasis - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulEmphasisAttr (ULONG) - input
Record attributes of the container.

Any or all of the following flags can be combined using the logical OR (|) operator:

CRA_CURSORED
Specifies that a record will be drawn with a selection cursor.

CRA_DISABLED
Specifies that a record will be drawn with unavailable-state emphasis.

CRA_INUSE
Specifies that a record will be drawn with in-use emphasis.

CRA_PICKED
Specifies that the record will be drawn with "picked" emphasis; for example, it is part of a lazy drag set.

CRA_SELECTED
Specifies that a record will be drawn with selected-state emphasis.

CRA_SOURCE
Specifies that a record will be drawn with source-menu emphasis.

fTurnOn (BOOL) - input
Flag indicating whether the specified attributes are to be set or reset.

TRUE	Set the specified attributes.
FALSE	Reset the specified attributes.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpCnrSetEmphasis - Remarks

This method provides a way to change the visual emphasis of an object. Visual emphasis can be used to give the user feedback as to the state of the object. For example, when a folder is opened on the desktop, in-use emphasis is added to the folder icon. The Workplace Shell accomplishes this by calling wpCnrSetEmphasis on the object.

This method is called automatically to set and reset the CRA_INUSE emphasis state during the processing of the wpAddToObjUseList and wpDeleteFromObjUseList methods, respectively.

This method is valid only for objects inserted into a container control using wpCnrInsertObject wpclsInsertMultipleObjects.

This method sends CM_SETRECORDEMPHASIS to the container. The CRA_* flags specified for the *ulEmphasisAttr* parameter are the same as those used for CM_SETRECORDEMPHASIS

wpCnrSetEmphasis - Usage

This method can be called at any time in order to set the emphasis for an object in an open folder. Some restrictions apply as to which CRA_* flags can be used.

wpCnrSetEmphasis - How to Override

This method is generally not overridden.

wpCnrSetEmphasis - Related Methods

Related Methods

- wpAddToObjUseList
- wpCnrInsertObject
- wpOpen

wpCnrSetEmphasis - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpConfirmDelete

wpConfirmDelete - Syntax

This instance method is called to allow the object to prompt the user to confirm the deletion of this object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
ULONG         fConfirmations;  /*  Confirmation flags. */
```

```
ULONG         rc;              /*  User-response value. */

rc = _wpConfirmDelete(somSelf, fConfirmations);
```

wpConfirmDelete Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpConfirmDelete Parameter - fConfirmations

fConfirmations (ULONG) - input
Confirmation flags.

CONFIRM_DELETE
Confirm for delete of all objects.
CONFIRM_DELETEFOLDER
Confirm for delete of folder objects only.

wpConfirmDelete Return Value - rc

rc (ULONG) - returns
User-response value.

The default processing returns one of the following:

OK_DELETE
Deletion of this object is confirmed.
NO_DELETE
Deletion of this object is cancelled.
CANCEL_DELETE
Deletion process from that point on is cancelled.

wpConfirmDelete - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

fConfirmations (ULONG) - input
Confirmation flags.

CONFIRM_DELETE
Confirm for delete of all objects.

CONFIRM_DELETEFOLDER
Confirm for delete of folder objects only.

rc (ULONG) - returns
User-response value.

The default processing returns one of the following:

OK_DELETE
Deletion of this object is confirmed.
NO_DELETE
Deletion of this object is cancelled.
CANCEL_DELETE
Deletion process from that point on is cancelled.

wpConfirmDelete - Remarks

This method is called during the processing of the wpDelete method only if the system confirm on delete flag is set.

wpConfirmDelete - Usage

This method is generally called only by the system.

wpConfirmDelete - How to Override

This method should be overridden in order to remove or replace the user prompt to confirm the delete operation.

wpConfirmDelete - Related Methods

Related Methods

- wpDelete
- wpFree

wpConfirmDelete - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpConfirmKeepAssoc

wpConfirmKeepAssoc - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method determines whether you want to rename the given file with different extensions and warns that associations might need to be fixed.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
ULONG             ulRename;  /*  Flag indicating whether the user wants to rename the file. */

ulRename = _wpConfirmKeepAssoc(somSelf);
```

wpConfirmKeepAssoc Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpConfirmKeepAssoc Return Value - ulRename

ulRename (ULONG) - returns
Flag indicating whether the user wants to rename the file.

Possible values are described in the following list:

KEEP_RENAMEFILESWITHEXT
The file is renamed, and any associations are added to the menu page.

DISCARD_RENAMEFILESWITHEXT
The file is renamed, and old associations are discarded (not added to the menu page).

CANCEL_RENAMEFILESWITHEXT
The file is not renamed, and the title is not changed.

wpConfirmKeepAssoc - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

ulRename (ULONG) - returns
Flag indicating whether the user wants to rename the file.

Possible values are described in the following list:

KEEP_RENAMEFILESWITHEXT
The file is renamed, and any associations are added to the menu page.

DISCARD_RENAMEFILESWITHEXT
The file is renamed, and old associations are discarded (not added to the menu page).

CANCEL_RENAMEFILESWITHEXT
The file is not renamed, and the title is not changed.

wpConfirmKeepAssoc - Usage

This method can be called only when confirmations are on and the extensions are different. It may be called by any class that wants to handle the prompting or overriding of return values.

wpConfirmKeepAssoc - Related Methods

Related Methods

- wpConfirmRenameFileWithExt
- wpSetTitleAndRenameFile

wpConfirmKeepAssoc - Topics

Select an item:
Syntax
Parameters
Returns
Usage
Related Methods
Glossary

wpConfirmObjectTitle

wpConfirmObjectTitle - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method displays the Object Title Confirmation dialog.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject      *somSelf;      /*  Pointer to the object on which the method is being invoked. */
WPFolder      *Folder;       /*  A pointer to the folder involved in the operation. */
WPObject     **ppDuplicate;  /*  A pointer to the address of an object with the same title in the folder. */
PSZ            pszTitle;     /*  The initial or returned name. */
ULONG          cbTitle;      /*  The length of the name's buffer (pszTitle). */
ULONG          menuID;       /*  The ID of the operation being performed. */
ULONG          ulNameClash;  /*  A flag indicating the action taken when two objects have the same title. */

ulNameClash = _wpConfirmObjectTitle(somSelf,
                Folder, ppDuplicate, pszTitle,
                cbTitle, menuID);
```

wpConfirmObjectTitle Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpConfirmObjectTitle Parameter - Folder

Folder (WPFolder *) - input
A pointer to the folder involved in the operation.

wpConfirmObjectTitle Parameter - ppDuplicate

ppDuplicate (WPObject **) - input
A pointer to the address of an object with the same title in the folder.

wpConfirmObjectTitle Parameter - pszTitle

pszTitle (PSZ) - in/out
The initial or returned name.

wpConfirmObjectTitle Parameter - cbTitle

cbTitle (ULONG) - input
The length of the name's buffer (*pszTitle*).

wpConfirmObjectTitle Parameter - menuID

menuID (ULONG) - input
The ID of the operation being performed.

For a listing of WPMENUIDs, see the individual object classes.

wpConfirmObjectTitle Return Value - ulNameClash

ulNameClash (ULONG) - returns
A flag indicating the action taken when two objects have the same title.

Possible values are described in the following list:

NAMECLASH_APPEND
The returned duplicate is appended to.
NAMECLASH_NONE
No object in the target folder has the same name.
NAMECLASH_NONE_KEEPASSOC
The file that was not renamed keeps the same associations.
NAMECLASH_RENAME
The file is renamed.
NAMECLASH_RENAME_KEEPASSOC
The renamed file keeps the same associations.
NAMECLASH_REPLACE
The returned duplicate is replaced.

wpConfirmObjectTitle - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

Folder (WPFolder *) - input
A pointer to the folder involved in the operation.

ppDuplicate (WPObject **) - input
A pointer to the address of an object with the same title in the folder.

pszTitle (PSZ) - in/out
The initial or returned name.

cbTitle (ULONG) - input
The length of the name's buffer (*pszTitle*).

menuID (ULONG) - input
The ID of the operation being performed.

For a listing of WPMENUIDs, see the individual object classes.

ulNameClash (ULONG) - returns
A flag indicating the action taken when two objects have the same title.

Possible values are described in the following list:

NAMECLASH_APPEND
The returned duplicate is appended to.
NAMECLASH_NONE
No object in the target folder has the same name.
NAMECLASH_NONE_KEEPASSOC
The file that was not renamed keeps the same associations.
NAMECLASH_RENAME
The file is renamed.
NAMECLASH_RENAME_KEEPASSOC
The renamed file keeps the same associations.
NAMECLASH_REPLACE
The returned duplicate is replaced.

wpConfirmObjectTitle - How to Override

This method is generally not overridden.

wpConfirmObjectTitle - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Glossary

wpConfirmRenameFileWithExt

wpConfirmRenameFileWithExt - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method prompts the user whether to rename this file with possibly a different extension.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somself;        /*  Pointer to the object invoking the method (a "self" pointer). */
ULONG             fConfirmation;  /*  Confirmation flag. */

fConfirmation = _wpConfirmRenameFileWithExt(
                somself);
```

wpConfirmRenameFileWithExt Parameter - somself

somself (WPFileSystem *) - input
Pointer to the object invoking the method (a "self" pointer).

Points to an object of class WPFileSystem.

wpConfirmRenameFileWithExt Return Value - fConfirmation

fConfirmation (ULONG) - returns
Confirmation flag.

CANCEL_RENAMEFILESWITHEXT
The file is not renamed, and the title is not changed.

DISCARD_RENAMEFILESWITHEXT
The file is renamed, and the old extension is discarded.

KEEP_RENAMEFILESWITHEXT
The file is renamed, but the old extension is kept.

wpConfirmRenameFileWithExt - Parameters

somself (WPFileSystem *) - input
Pointer to the object invoking the method (a "self" pointer).

Points to an object of class WPFileSystem.

fConfirmation (ULONG) - returns
Confirmation flag.

CANCEL_RENAMEFILESWITHEXT
The file is not renamed, and the title is not changed.

DISCARD_RENAMEFILESWITHEXT
The file is renamed, and the old extension is discarded.

KEEP_RENAMEFILESWITHEXT
The file is renamed, but the old extension is kept.

wpConfirmRenameFileWithExt - Remarks

This method is called when confirmations are set for wpSetTitleAndRenameFile and the extensions are different.

wpConfirmRenameFileWithExt - Usage

This method can be called at any time by a class that wants to handle the prompting or override the return value.

wpConfirmRenameFileWithExt - How to Override

This method is generally not overridden.

wpConfirmRenameFileWithExt - Related Methods

Related Methods

- wpSetTitleAndRenameFile

wpConfirmRenameFileWithExt - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpContainsFolders

wpContainsFolders - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method determines whether the folder contains any subfolders.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
BOOL         *pfSubFolders;  /*  Flag indicating whether the folder contains subfolders. */
BOOL          rc;            /*  Success indicator. */

rc = _wpContainsFolders(somSelf, pfSubFolders);
```

wpContainsFolders Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpContainsFolders Parameter - pfSubFolders

pfSubFolders (BOOL *) - output
Flag indicating whether the folder contains subfolders.

Possible values are described in the following list:

TRUE
The folder contains subfolders.
FALSE
The folder does not contain subfolders.

wpContainsFolders Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpContainsFolders - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

pfSubFolders (BOOL *) - output
Flag indicating whether the folder contains subfolders.

Possible values are described in the following list:

TRUE
The folder contains subfolders.
FALSE
The folder does not contain subfolders.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpContainsFolders - How to Override

This method is generally not overridden.

wpContainsFolders - Related Methods

Related Methods

- wpclsQueryOpenFolders

wpContainsFolders - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpCopiedFromTemplate

wpCopiedFromTemplate - Syntax

This instance method is called to allow an object to perform class-specific processing when a new object is created from a template.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */

wpCopiedFromTemplate(somSelf);
```

wpCopiedFromTemplate Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpCopiedFromTemplate - Return Value

There is no return value for this method.

wpCopiedFromTemplate - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

There is no return value for this method.

wpCopiedFromTemplate - Remarks

This method is called immediately after a new object is created from a template. The system does not do any default processing for this method. This method is typically overridden to perform class-specific initialization on a created object. For example, a customer order-form class would prefill the date, time, and order number in an instance of itself that was created from a template.

wpCopiedFromTemplate - Usage

This method is generally called only by the system.

wpCopiedFromTemplate - How to Override

This method should be overridden by object classes that need to initialize the new object after it is created from a template.

wpCopiedFromTemplate - Topics

Select an item:
Syntax
Parameters
Returns

Remarks
How to Override
Usage
Glossary

wpCopyObject

wpCopyObject - Syntax

This instance method is called to create a new copy of the object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
WPFolder     *Folder;   /*  Pointer to a Folder object. */
BOOL          fLock;    /*  Lock object flag. */
WPObject     *rc;       /*  Pointer to the newly created object. */

rc = _wpCopyObject(somSelf, Folder, fLock);
```

wpCopyObject Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpCopyObject Parameter - Folder

Folder (WPFolder *) - input
Pointer to a Folder object.

A pointer to a Folder object in which to place this new object. This pointer can be determined by issuing a call to the wpclsQueryFolder method.

wpCopyObject Parameter - fLock

fLock (BOOL) - input
Lock object flag.

If this flag is false, the newly created object will be made dormant whenever the object and the folder containing the object are closed. If this flag is true, the new object will remain active until the caller issues the wpUnlockObject method on it.

wpCopyObject Return Value - rc

rc (WPObject *) - returns
Pointer to the newly created object.

A return value of NULL indicates that an error occurred.

wpCopyObject - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

Folder (WPFolder *) - input
Pointer to a Folder object.

A pointer to a Folder object in which to place this new object. This pointer can be determined by issuing a call to the wpclsQueryFolder method.

fLock (BOOL) - input
Lock object flag.

If this flag is false, the newly created object will be made dormant whenever the object and the folder containing the object are closed. If this flag is true, the new object will remain active until the caller issues the wpUnlockObject method on it.

rc (WPObject *) - returns
Pointer to the newly created object.

A return value of NULL indicates that an error occurred.

wpCopyObject - Remarks

Copies of an object can always be deleted and moved by default, even if the original has the OBJSTYLE_NODELETE or OBJSTYLE_NOMOVE style set.

wpCopyObject - Usage

This method can be called at any time in order to create a copy of an existing object.

wpCopyObject - How to Override

This method can be overridden by classes which need to keep track of where instances are generated.

wpCopyObject - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Glossary

wpCreateAnother

wpCreateAnother - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to create another object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
PSZ           pszTitle;     /*  String containing the title of the object. */
PSZ           pszSetupEnv;  /*  Environment setup string. */
WPFolder     *Folder;       /*  Pointer to the container. */
WPObject     *Object;       /*  Pointer to the new object. */

Object = _wpCreateAnother(somSelf, pszTitle,
           pszSetupEnv, Folder);
```

wpCreateAnother Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpCreateAnother Parameter - pszTitle

pszTitle (PSZ) - input
String containing the title of the object.

wpCreateAnother Parameter - pszSetupEnv

pszSetupEnv (PSZ) - input
Environment setup string.

This parameter is a super-string containing parameters to the new object that are extracted when the wpSetup method is called on that object.

wpCreateAnother Parameter - Folder

Folder (WPFolder *) - input
Pointer to the container.

wpCreateAnother Return Value - Object

Object (WPObject *) - returns
Pointer to the new object.

wpCreateAnother - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pszTitle (PSZ) - input
String containing the title of the object.

pszSetupEnv (PSZ) - input
Environment setup string.

This parameter is a super-string containing parameters to the new object that are extracted when the wpSetup method is called on that object.

Folder (WPFolder *) - input
Pointer to the container.

Object (WPObject *) - returns
Pointer to the new object.

wpCreateAnother - Remarks

This instance method gives a subclass the opportunity to return an object of its own choosing; for example, WPProgramFile returns an instance of WPProgram.

wpCreateAnother - How to Override

This method can be overridden only in a subclass.

wpCreateAnother - Related Methods

Related Methods

- wpCreateFromTemplate
- wpCreateShadowObject

wpCreateAnother - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpCreateFromTemplate

wpCreateFromTemplate - Syntax

This instance method is called to create an object from a template.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
WPFolder     *folder;   /*  Pointer to a Folder object. */
BOOL          fLock;    /*  Lock object flag. */
WPObject     *rc;       /*  Pointer to the new object. */

rc = _wpCreateFromTemplate(somSelf, folder,
       fLock);
```

wpCreateFromTemplate Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpCreateFromTemplate Parameter - folder

folder (WPFolder *) - input
Pointer to a Folder object.

A pointer to a Folder object in which to place the new object. This pointer can be determined by issuing a call to the wpclsQueryFolder method.

wpCreateFromTemplate Parameter - fLock

fLock (BOOL) - input
Lock object flag.

If this flag is false, the newly created object will be made dormant whenever the object and the folder containing the object are closed. If this flag is true, the new flag will remain active until the caller issues the wpUnlockObject method on it.

wpCreateFromTemplate Return Value - rc

rc (WPObject *) - returns
Pointer to the new object.

A return value of NULL indicates that an error occurred.

wpCreateFromTemplate - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

folder (WPFolder *) - input
Pointer to a Folder object.

A pointer to a Folder object in which to place the new object. This pointer can be determined by issuing a call to the wpclsQueryFolder

method.

fLock (BOOL) - input
Lock object flag.

If this flag is false, the newly created object will be made dormant whenever the object and the folder containing the object are closed.
If this flag is true, the new flag will remain active until the caller issues the wpUnlockObject method on it.

rc (WPObject *) - returns
Pointer to the new object.

A return value of NULL indicates that an error occurred.

wpCreateFromTemplate - Remarks

The new object will be an identical copy of the template object except that the OBJSTYLE_TEMPLATE object style will be taken out. wpCopiedFromTemplate will be called on the new object.

wpCreateFromTemplate - Usage

This method can be called at any time in order to create a new object from a template object.

wpCreateFromTemplate - How to Override

This method is generally not overridden.

wpCreateFromTemplate - Related Methods

Related Methods

- wpCopyObject
- wpclsNew

wpCreateFromTemplate - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpCreateShadowObject

wpCreateShadowObject - Syntax

This instance method is called to create a shadow of an object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
WPFolder     *Folder;   /*  Pointer to a Folder object. */
BOOL          fLock;    /*  Lock object flag. */
WPObject     *rc;       /*  Pointer to the new shadow object. */

rc = _wpCreateShadowObject(somSelf, Folder,
       fLock);
```

wpCreateShadowObject Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpCreateShadowObject Parameter - Folder

Folder (WPFolder *) - input
Pointer to a Folder object.

A pointer to a Folder object in which to place the new shadow object. This pointer can be determined by issuing a call to the wpclsQueryFolder method.

wpCreateShadowObject Parameter - fLock

fLock (BOOL) - input
Lock object flag.

If this flag is false, the new shadow object will be made dormant whenever the object and the folder containing the object are closed.

If this flag is true, the new object will remain awake until the caller issues the wpUnlockObject method on it.

wpCreateShadowObject Return Value - rc

rc (WPObject *) - returns
Pointer to the new shadow object.

A return value of NULL indicates that an error occurred.

wpCreateShadowObject - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

Folder (WPFolder *) - input
Pointer to a Folder object.

A pointer to a Folder object in which to place the new shadow object. This pointer can be determined by issuing a call to the wpclsQueryFolder method.

fLock (BOOL) - input
Lock object flag.

If this flag is false, the new shadow object will be made dormant whenever the object and the folder containing the object are closed.

If this flag is true, the new object will remain awake until the caller issues the wpUnlockObject method on it.

rc (WPObject *) - returns
Pointer to the new shadow object.

A return value of NULL indicates that an error occurred.

wpCreateShadowObject - Remarks

The new object is created as an instance of class WPShadow.

wpCreateShadowObject - Usage

This method can be called at any time in order to create a shadow object for this object.

wpCreateShadowObject - How to Override

This method is generally not overridden.

wpCreateShadowObject - Example Code

This example creates a shadow of the Desktop.

```
#define INCL_WINWORKPLACE
#include <os2.h>

SOMAny    *objDesktop;
SOMAny    *objSysShadow;
     /* Get the object pointer of the desktop folder */

objDesktop = _wpclsQueryFolder( _WPFolder
                              , "<WP_DESKTOP>"     /* Folder location */
                              ,TRUE);              /* Keep active */

     /* Create a shadow of the desktop */

objSysShadow = _wpCreateShadowObject( self
                                    , objDesktop   /* Object to shadow */
                                    ,FALSE);       /* Make dormant when folder
                                                      closes */
```

wpCreateShadowObject - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Glossary

wpCreateShadowObjectExt

wpCreateShadowObjectExt - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to create a shadow of an object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject       *somSelf;      /*  Pointer to the object on which the method is being invoked. */
WPFolder       *Folder;       /*  Pointer to a Folder object. */
BOOL            fLock;        /*  Lock object flag. */
PSZ             pszSetup;     /*  Pointer to a setup string. */
M_WPObject     *shadowClass;  /*  Class of the new shadow object. */
WPObject       *Object;       /*  Pointer to the new shadow object. */

Object = _wpCreateShadowObjectExt(somSelf,
           Folder, fLock, pszSetup, shadowClass);
```

wpCreateShadowObjectExt Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpCreateShadowObjectExt Parameter - Folder

Folder (WPFolder *) - input
Pointer to a Folder object.

A pointer to a Folder object in which to place the new shadow object. This pointer can be determined by issuing a call to the wpclsQueryFolder method.

wpCreateShadowObjectExt Parameter - fLock

fLock (BOOL) - input
Lock object flag.

If this flag is false, the new shadow object will be made dormant whenever the object and the folder containing the object are closed.

If this flag is true, the new object will remain awake until the caller issues the wpUnlockObject method on it.

wpCreateShadowObjectExt Parameter - pszSetup

pszSetup (PSZ) - input
Pointer to a setup string.

wpCreateShadowObjectExt Parameter - shadowClass

shadowClass (M_WPObject *) - input
Class of the new shadow object.

wpCreateShadowObjectExt Return Value - Object

Object (WPObject *) - returns
Pointer to the new shadow object.

A value of NULL indicates that an error occurred.

wpCreateShadowObjectExt - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

Folder (WPFolder *) - input
Pointer to a Folder object.

A pointer to a Folder object in which to place the new shadow object. This pointer can be determined by issuing a call to the wpclsQueryFolder method.

fLock (BOOL) - input
Lock object flag.

If this flag is false, the new shadow object will be made dormant whenever the object and the folder containing the object are closed.

If this flag is true, the new object will remain awake until the caller issues the wpUnlockObject method on it.

pszSetup (PSZ) - input
Pointer to a setup string.

shadowClass (M_WPObject *) - input
Class of the new shadow object.

Object (WPObject *) - returns
Pointer to the new shadow object.

A value of NULL indicates that an error occurred.

wpCreateShadowObjectExt - Remarks

This method is used instead of wpCreateShadowObject when setup string needs to be passed to the new shadow object or when the shadow is to be created as a subclass of a shadow.

wpCreateShadowObjectExt - Usage

This method can be called at any time in order to create a shadow object for this object.

wpCreateShadowObjectExt - How to Override

This method is generally not overridden.

wpCreateShadowObjectExt - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Glossary

wpDelete

wpDelete - Syntax

This instance method is called to delete an object and prompt for confirmation if necessary.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
ULONG         fConfirmations;  /*  Confirmation flags. */
ULONG         rc;              /*  Success indicator. */

rc = _wpDelete(somSelf, fConfirmations);
```

wpDelete Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpDelete Parameter - fConfirmations

fConfirmations (ULONG) - input
Confirmation flags.

CONFIRM_DELETE
Prompt for confirmation for all objects.
CONFIRM_DELETEFOLDER
Prompt for confirmation for folder objects, only.
NULL

No confirmations.

wpDelete Return Value - rc

rc (ULONG) - returns
Success indicator.

NO_DELETE	Error occurred.
CANCEL_DELETE	User canceled the operation.
OK_DELETE	Object was deleted.

wpDelete - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

fConfirmations (ULONG) - input
Confirmation flags.

CONFIRM_DELETE
Prompt for confirmation for all objects.
CONFIRM_DELETEFOLDER
Prompt for confirmation for folder objects, only.
NULL
No confirmations.

rc (ULONG) - returns
Success indicator.

NO_DELETE	Error occurred.
CANCEL_DELETE	User canceled the operation.
OK_DELETE	Object was deleted.

wpDelete - Remarks

The confirmation flags are passed to the wpConfirmDelete method. If wpConfirmDelete returns OK_DELETE, the wpFree method is called on the object.

wpDelete - Usage

This method can be called at any time in order to delete an object.

wpDelete - How to Override

This method is generally not overridden.

wpDelete - Related Methods

Related Methods

- wpConfirmDelete
- wpDelete
- wpFree
- wpQueryConfirmations

wpDelete - Example Code

This example does special processing upon opening a folder.

```
SOM_Scope HWND   SOMLINK myfold_wpOpen(MYFOLDER *somSelf,
                 HWND hwndCnr,
                 ULONG ulView,
                 ULONG param)

{
    HWND        hwndFromParentOpen = NULLHANDLE;
    MYFOLDERData *somThis = MYFOLDERGetData(somSelf);
    MYFOLDERMethodDebug("MYFOLDER","myfold_wpOpen");

    if (_queryFirstViewHandle(somSelf) == NULLHANDLE)
    {
              /* ONLY do this on FIRST open of the folder. */

       if (_wpPopulate(somSelf, NULLHANDLE, NULL, FALSE))
        {
           WPObject *Obj;
           WPObject *LastFoundObj = NULL;

           /* Check contents of folder and see which are instances of
            * WPFolder, we'll delete any WPFolders as a cleanup step */

           for ( Obj = _wpQueryContent(somSelf,NULL,(ULONG)QC_First);
                 Obj;
                 Obj = _wpQueryContent(somSelf, Obj, (ULONG) QC_Next ))

           {
              /* delete the last object found on the previous iteration
               * of the loop (we couldn't delete it then, because we
               * needed it at the top of the loop for this iteration)
               */
              if (LastFoundObj) {
                 _wpDelete(LastFoundObj,0);
                 LastFoundObj = NULL;
              } /* endif */

              /* we want to make sure that all previously
               * existing folders are deleted */

              if (_somIsA(Obj, _WPFolder)) {
                 LastFoundObj = Obj;
              }  /* endif */

           }  /* endfor */

           /* if there is still one object left to delete, do it now */


           if (LastFoundObj) {
              _wpDelete(LastFoundObj,0);
              LastFoundObj = NULL;
           } /* endif */
```

```
      } /* end if populate */
   } /* end if first open */

            /* Show the opened folder */

   hwndFromParentOpen = parent_wpOpen(somSelf, hwndCnr, ulView, param);

            /* If this is the first open, set our global flag */

   if (_queryFirstViewHandle(somSelf) == NULLHANDLE)  {
      _setFirstViewHandle(somSelf, hwndFromParentOpen);
   } /* endif */

   return( hwndFromParentOpen );
}
```

wpDelete - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpDeleteAllJobs

wpDeleteAllJobs - Syntax

This instance method is called to delete all jobs in a printer (print destination) object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPrinter     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL           rc;       /*  Success indicator. */

rc = _wpDeleteAllJobs(somSelf);
```

wpDeleteAllJobs Parameter - somSelf

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

wpDeleteAllJobs Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpDeleteAllJobs - Parameters

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpDeleteAllJobs - How to Override

This method is generally not overridden.

wpDeleteAllJobs - Related Methods

Related Methods

- wpDeleteJob

wpDeleteAllJobs - Example Code

```
VOID Abort_All_Jobs( SOMAny *self )

{

   CHAR   szTitle[256];           /* Title of printer object */

   strcpy( szTitle, _wpQueryTitle( self ) );
```

```
    somPrintf("%s: Deleting all jobs.\n", szTitle);

    _wpSetDefaultPrinter( self );

    _wpHoldPrinter( self );

    _wpDeleteAllJobs( self );

}
```

wpDeleteAllJobs - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Example Code
Related Methods
Glossary

wpDeleteContents

wpDeleteContents - Syntax

This instance method is called to delete the contents of a folder.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
ULONG         fConfirmations;  /*  Confirmation flags. */
ULONG         rc;              /*  Success indicator. */

rc = _wpDeleteContents(somSelf, fConfirmations);
```

wpDeleteContents Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpDeleteContents Parameter - fConfirmations

fConfirmations (ULONG) - input
Confirmation flags.

CONFIRM_DELETE	Prompt for confirmation for all objects.
CONFIRM_DELETEFOLDER	Prompt for confirmation for folder objects only.
NULL	No confirmations.

wpDeleteContents Return Value - rc

rc (ULONG) - returns
Success indicator.

OK_DELETE	All objects were deleted.
NO_DELETE	Error occurred. At least one object was not deleted.
CANCEL_DELETE	User canceled the operation.

wpDeleteContents - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

fConfirmations (ULONG) - input
Confirmation flags.

CONFIRM_DELETE	Prompt for confirmation for all objects.
CONFIRM_DELETEFOLDER	Prompt for confirmation for folder objects only.
NULL	No confirmations.

rc (ULONG) - returns
Success indicator.

OK_DELETE	All objects were deleted.
NO_DELETE	Error occurred. At least one object was not deleted.
CANCEL_DELETE	User canceled the operation.

wpDeleteContents - Remarks

wpDelete and wpFree will automatically call this instance method for folder objects.

wpDeleteContents - Usage

This method can be called at any time in order to delete the contents of a folder.

wpDeleteContents - How to Override

This method is generally not overridden.

wpDeleteContents - Related Methods

Related Methods

- wpFree
- wpDelete

wpDeleteContents - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpDeleteFromContent

wpDeleteFromContent - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method removes an object from a folder's Content List.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
WPObject     *Object;   /*  Pointer to the object to be deleted from the folder's content list. */
BOOL          rc;       /*  Success indicator. */

rc = _wpDeleteFromContent(somSelf, Object);
```

wpDeleteFromContent Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpDeleteFromContent Parameter - Object

Object (WPObject *) - input
Pointer to the object to be deleted from the folder's content list.

wpDeleteFromContent Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpDeleteFromContent - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

Object (WPObject *) - input
Pointer to the object to be deleted from the folder's content list.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpDeleteFromContent - How to Override

This method should be overridden to intercept the notification of the "Removed an object from a folder" event for subclasses that define their own folder views. The parent must always be called.

wpDeleteFromContent - Related Methods

Related Methods

- wpAddToContent

wpDeleteFromContent - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpDeleteFromObjUseList

wpDeleteFromObjUseList - Syntax

This instance method is called to remove an item type from an object's in-use list.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
PUSEITEM      pUseItem;  /*  Pointer to a USEITEM structure. */
BOOL          rc;        /*  Success indicator */

rc = _wpDeleteFromObjUseList(somSelf, pUseItem);
```

wpDeleteFromObjUseList Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpDeleteFromObjUseList Parameter - pUseItem

pUseItem (PUSEITEM) - input
Pointer to a USEITEM structure.

wpDeleteFromObjUseList Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpDeleteFromObjUseList - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pUseItem (PUSEITEM) - input
Pointer to a USEITEM structure.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpDeleteFromObjUseList - Remarks

This method will remove a specified item type from an object's in-use (USEITEM) list.

If the usage item being removed is of type *USAGE_OPENVIEW*, the in-use emphasis bit will be turned off for all inserted records for this object.

wpDeleteFromObjUseList - Usage

A call to this method should be made when a particular usage item, added to the in-use list via the wpAddToObjUseList method, is no longer needed.

This method must be called before the USEITEM memory is freed.

wpDeleteFromObjUseList - How to Override

This method is generally not overridden.

wpDeleteFromObjUseList - Related Methods

Related Methods

- wpAddToObjUseList
- wpFindUseItem

wpDeleteFromObjUseList - Example Code

In this example, when the last view of a MYFOLDER object is closed, we zero out its first view instance variable, so that the next time it has a first view opened, we can keep track of that.

```
SOM_Scope BOOL  SOMLINK myfold_wpDeleteFromObjUseList(MYFOLDER *somSelf,
                        PUSEITEM  pUseItem )
{
   MYFOLDERData *somThis   = MYFOLDERGetData(somSelf);
   PUSEITEM      pItem;
   BOOL          rcParentCall;
   MYFOLDERMethodDebug("MYFOLDER","myfold_wpDeleteFromObjUseList");

          /* Call parent's method first */

   rcParentCall = parent_wpDeleteFromObjUseList(somSelf, pUseItem);

   pItem = _wpFindUseItem(somSelf, USAGE_OPENVIEW, NULL);

   if (!pItem) {

      _setFirstViewHandle(somSelf, NULLHANDLE);

   } /* endif */

   return( rcParentCall );
}
```

wpDeleteFromObjUseList - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpDeleteJob

wpDeleteJob - Syntax

This instance method is called to delete a job object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPJob     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL       rc;       /*  Success indicator. */

rc = _wpDeleteJob(somSelf);
```

wpDeleteJob Parameter - somSelf

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

wpDeleteJob Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpDeleteJob - Parameters

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpDeleteJob - How to Override

This method is generally not overridden.

wpDeleteJob - Related Methods

Related Methods

- wpDeleteAllJobs

wpDeleteJob - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpDisplayHelp

wpDisplayHelp - Syntax

This instance method is called to allow the object to display a help panel.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
ULONG         HelpPanelId;  /*  Object's help panel ID. */
PSZ           HelpLibrary;  /*  Pointer to the help library. */
BOOL          rc;           /*  Success indicator. */

rc = _wpDisplayHelp(somSelf, HelpPanelId,
       HelpLibrary);
```

wpDisplayHelp Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpDisplayHelp Parameter - HelpPanelId

HelpPanelId (ULONG) - input
Object's help panel ID.

wpDisplayHelp Parameter - HelpLibrary

HelpLibrary (PSZ) - input
Pointer to the help library.

A pointer to a zero-terminated string which contains the name of the help library.

wpDisplayHelp Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpDisplayHelp - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

HelpPanelId (ULONG) - input
Object's help panel ID.

HelpLibrary (PSZ) - input
Pointer to the help library.

A pointer to a zero-terminated string which contains the name of the help library.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpDisplayHelp - Remarks

This method allows each object class to display a help panel in response to a users request for help. The help panel will be displayed by the shell on a help instance managed by the shell.

wpDisplayHelp - Usage

This method can be called at any time in order to display a help panel.

wpDisplayHelp - How to Override

This method is generally not overridden.

wpDisplayHelp - Related Methods

Related Methods

- wpDisplayHelp
- wpMenuItemHelpSelected
- wpQueryDefaultHelp
- wpSetDefaultHelp
- wpclsQueryDefaultHelp

wpDisplayHelp - Example Code

This example overrides wpMenuItemHelpSelected and provides help for the Hex and Text view popup menu selections.

```
SOM_Scope BOOL   SOMLINK Browse_O_Maticwps_wpMenuItemHelpSelected(Browse_O_Matic *somSelf,
                 ULONG MenuId)
{
    /* Browse_O_MaticData *somThis = Browse_O_MaticGetData(somSelf); */
    Browse_O_MaticMethodDebug("Browse_O_Matic","Browse_O_Maticwps_wpMenuItemHelpSelected");

    switch( MenuId){

      case ID_OPEN_TEXT:
         return( _wpDisplayHelp( somSelf, PANEL_HELP_OPEN_TEXT, "pmbrowse.hlp"));

      case ID_OPEN_HEX:
         return( _wpDisplayHelp( somSelf, PANEL_HELP_OPEN_HEX, "pmbrowse.hlp"));

      default:
         break;
    }

    return( FALSE);
}
```

wpDisplayHelp - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code

Related Methods
Glossary

wpDisplayMenu

wpDisplayMenu - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method creates and displays an object's popup context menu.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
HWND          hwndOwner;   /*  Handle of the window that the menu is owned by. */
HWND          hwndClient;  /*  Handle of the window on which the menu is displayed. */
POINTL       *ptlPopupPt;  /*  Coordinates of the starting position for the menu. */
ULONG         ulMenuType;  /*  Type of menu to display. */
ULONG         ulReserved;  /*  Reserved value. */
HWND          hwndMenu;    /*  Menu handle. */

hwndMenu = _wpDisplayMenu(somSelf, hwndOwner,
             hwndClient, ptlPopupPt, ulMenuType,
             ulReserved);
```

wpDisplayMenu Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpDisplayMenu Parameter - hwndOwner

hwndOwner (HWND) - input
Handle of the window that the menu is owned by.

This parameter is required, and must not be NULL.

wpDisplayMenu Parameter - hwndClient

hwndClient (HWND) - input
Handle of the window on which the menu is displayed.

This window must be owned by *hwndOwner* .

If *hwndClient* is supplied:

- The pop-up point is assumed to be in *hwndClient* window coordinates.
- *hwndClient* is given the focus before displaying the menu.

If *hwndClient* is not supplied:

- The pop-up point is assumed to be in *hwndOwner* window coordinates.
- *hwndOwner* is given the focus before displaying the menu.

wpDisplayMenu Parameter - ptlPopupPt

ptlPopupPt (POINTL *) - input
Coordinates of the starting position for the menu.

For MENU_OBJECTPOPUP requests where *hwndOwner* is an icon or tree view of a container which contains the object, this parameter is ignored, and the menu is displayed at the lower right-hand corder of the object's icon. This parameter is also ignored for MENU_TITLEBARPULLDOWNxx requests.

wpDisplayMenu Parameter - ulMenuType

ulMenuType (ULONG) - input
Type of menu to display.

Possible values are described in the following list:

MENU_EDITPULLDOWN
MENU_FOLDERMENUBAR
MENU_FOLDERPULLDOWN
MENU_HELPPULLDOWN
MENU_NODISPLAY
MENU_OBJECTPOPUP
MENU_OPENVIEWPOPUP
MENU_SELECTEDPULLDOWN
MENU_TITLEBARPULLDOWN
MENU_TITLEBARPULLDOWNINT
MENU_USER
MENU_VIEWPULLDOWN

wpDisplayMenu Parameter - ulReserved

ulReserved (ULONG) - input
Reserved value.

wpDisplayMenu Return Value - hwndMenu

hwndMenu (HWND) - returns
Menu handle.

A return value of NULLHANDLE indicates that either an invalid *ulMenuType* or an invalid *hwndClient* or *hwndOwner* was given.

wpDisplayMenu - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndOwner (HWND) - input
Handle of the window that the menu is owned by.

This parameter is required, and must not be NULL.

hwndClient (HWND) - input
Handle of the window on which the menu is displayed.

This window must be owned by *hwndOwner* .

If *hwndClient* is supplied:

- The pop-up point is assumed to be in *hwndClient* window coordinates.
- *hwndClient* is given the focus before displaying the menu.

If *hwndClient* is not supplied:

- The pop-up point is assumed to be in *hwndOwner* window coordinates.
- *hwndOwner* is given the focus before displaying the menu.

ptlPopupPt (POINTL *) - input
Coordinates of the starting position for the menu.

For MENU_OBJECTPOPUP requests where *hwndOwner* is an icon or tree view of a container which contains the object, this parameter is ignored, and the menu is displayed at the lower right-hand corder of the object's icon. This parameter is also ignored for MENU_TITLEBARPULLDOWNxx requests.

ulMenuType (ULONG) - input
Type of menu to display.

Possible values are described in the following list:

MENU_EDITPULLDOWN
MENU_FOLDERMENUBAR
MENU_FOLDERPULLDOWN
MENU_HELPPULLDOWN
MENU_NODISPLAY
MENU_OBJECTPOPUP
MENU_OPENVIEWPOPUP
MENU_SELECTEDPULLDOWN
MENU_TITLEBARPULLDOWN
MENU_TITLEBARPULLDOWNINT
MENU_USER
MENU_VIEWPULLDOWN

ulReserved (ULONG) - input
Reserved value.

hwndMenu (HWND) - returns
Menu handle.

A return value of NULLHANDLE indicates that either an invalid *ulMenuType* or an invalid *hwndClient* or *hwndOwner* was given.

wpDisplayMenu - Usage

This method can be called at any time.

wpDisplayMenu - How to Override

This method should not be overridden.

wpDisplayMenu - Related Methods

Related Methods

- wpInsertPopupMenuItems
- wpModifyPopupMenu

wpDisplayMenu - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpDoesObjectMatch

wpDoesObjectMatch - Syntax

This instance method is called to allow the object to determine if it matches the specified criteria.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;                /*  Pointer to the object on which the method is being invoked. */
PVOID         pvoidExtendedCriteria;  /*  Pointer to class-specific extended search criteria. */
BOOL          rc;                     /*  Success indicator. */

rc = _wpDoesObjectMatch(somSelf, pvoidExtendedCriteria);
```

wpDoesObjectMatch Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpDoesObjectMatch Parameter - pvoidExtendedCriteria

pvoidExtendedCriteria (PVOID) - input
Pointer to class-specific extended search criteria.

A pointer to a buffer that contains the class-specific extended search criteria.

wpDoesObjectMatch Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	The object matches the specified criteria.
FALSE	The object does not match the specified criteria.

wpDoesObjectMatch - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pvoidExtendedCriteria (PVOID) - input
Pointer to class-specific extended search criteria.

A pointer to a buffer that contains the class-specific extended search criteria.

rc (BOOL) - returns
Success indicator.

TRUE	The object matches the specified criteria.
FALSE	The object does not match the specified criteria.

wpDoesObjectMatch - Remarks

If the object class has extended search criteria, this method is called to determine if the object found matches the extended search criteria.

wpDoesObjectMatch - Usage

This method can be called at any time in order to determine if an object matches some extended search criteria.

wpDoesObjectMatch - How to Override

This method should be overridden by classes which introduce extended search criteria for use by the Find and Include facilities.

wpDoesObjectMatch - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Glossary

wpDragCell

wpDragCell - Syntax

This instance method is called to drag-apply a value in a cell.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPalette     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PCELL          pCell;    /*  Pointer to the CELL structure to be dragged. */
HWND           hwndPal;  /*  Palette window handle. */
PPOINTL        ptlDrag;  /*  Pointer in hwndPal coordinates from which drag was initiated. */
BOOL           rc;       /*  Success indicator. */

rc = _wpDragCell(somSelf, pCell, hwndPal,
       ptlDrag);
```

wpDragCell Parameter - somSelf

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

wpDragCell Parameter - pCell

pCell (PCELL) - input
Pointer to the CELL structure to be dragged.

wpDragCell Parameter - hwndPal

hwndPal (HWND) - input
Palette window handle.

wpDragCell Parameter - ptlDrag

ptlDrag (PPOINTL) - input
Pointer in *hwndPal* coordinates from which drag was initiated.

wpDragCell Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpDragCell - Parameters

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

pCell (PCELL) - input
Pointer to the CELL structure to be dragged.

hwndPal (HWND) - input
Palette window handle.

ptlDrag (PPOINTL) - input
Pointer in *hwndPal* coordinates from which drag was initiated.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpDragCell - Remarks

The default processing for this method by the WPPalette class is to do nothing other than return FALSE.

wpDragCell - Usage

This method is generally called only by the palette window after it receives the WM_BEGINDRAG message.

wpDragCell - How to Override

This method should be overridden to handle the drag-apply action. Override processing should include capturing the mouse and waiting for and processing the WM_ENDDRAG message.

wpDragCell - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Glossary

wpDraggedOverObject

wpDraggedOverObject - Syntax

This instance method may be called on an object that is currently being dragged with the mouse to identify the current target object. The

return code from this method lets the system know whether the object being dragged can be dropped on the specified target.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;            /*  Pointer to the object on which the method is being invoked. */
WPObject     *DraggedOverObject;  /*  Pointer to the object that the drag cursor is over. */
MRESULT       rc;                 /*  Return code. */

rc = _wpDraggedOverObject(somSelf, DraggedOverObject);
```

wpDraggedOverObject Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpDraggedOverObject Parameter - DraggedOverObject

DraggedOverObject (WPObject *) - input
Pointer to the object that the drag cursor is over.

The current target object for the drag operation.

Points to an object of class WPObject.

wpDraggedOverObject Return Value - rc

rc (MRESULT) - returns
Return code.

DropIndicator (USHORT)
Low-word.

This code is one of the DOR_* constants, such as DOR_DROP or DOR_NODROP, which indicate whether a drop is allowed on the current target.

DropOperation (USHORT)
High-word.

The current drag operation code. Examples are DO_COPY, DO_MOVE or DO_LINK to indicate that the drag action over this target should be a copy, move, or a link.

wpDraggedOverObject - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

DraggedOverObject (WPObject *) - input
Pointer to the object that the drag cursor is over.

The current target object for the drag operation.

Points to an object of class WPObject.

rc (MRESULT) - returns
Return code.

DropIndicator (USHORT)
Low-word.

This code is one of the DOR_* constants, such as DOR_DROP or DOR_NODROP, which indicate whether a drop is allowed on the current target.

DropOperation (USHORT)
High-word.

The current drag operation code. Examples are DO_COPY, DO_MOVE or DO_LINK to indicate that the drag action over this target should be a copy, move, or a link.

wpDraggedOverObject - Remarks

When a target object is dragged over by the mouse, it will always receive a wpDragOver instance method call. Many target objects will choose to decide the current drag operation and whether a drop is possible based upon their own rules. For instance, the WPShredder object will return DO_DROP,DO_MOVE if it decides that all the source objects can be deleted. However, some targets require the source or sources to participate in the decision over whether they can accept the drop. The way that a target allows a source object to have a say in what the drop action will be is by calling the wpDraggedOverObject on each source object. The wpDraggedOverObject instance method may be invoked, on an object that is being dragged (source object) at any time, to see if it can support a drop on the current target. If the object that is being dragged responds favorably to this method, it may later receive a wpDroppedOnObject instance method call so that it can process the drop action.

As an example, consider the case where a program object is dragged onto a data file object. The program would respond DO_DROP to the wpDraggedOverObject instance method, so that the data file would be a valid drop target. If the user chose to allow the drop, then the program would receive a wpDroppedOnObject instance method. Then the program would be able to open itself as a viewer of the data file object.

This method is called as a result of a DM_DRAGOVER message being sent. For further documentation of the possible return values, see DM_DRAGOVER in the *Presentation Manager Programming Reference* .

wpDraggedOverObject - Usage

This method is typically called by objects that require participation from the source object when a drop occurs. The method can be called at any time; however, the method would normally be called only by a target object on one of the source objects during a drag or drop operation.

wpDraggedOverObject - How to Override

Override this method if your object class wishes to allow itself to be used as a source object that can perform a drop operation. A favorable return code from this method may lead to a wpDroppedOnObject instance method being invoked on the source object. The instance method would be overridden to actually do the drop operation.

wpDraggedOverObject - Related Methods

Related Methods

- wpDragOver
- wpDrop
- wpDroppedOnObject
- wpFormatDragItem

wpDraggedOverObject - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpDragOver

wpDragOver - Syntax

This instance method is called to inform the object that other objects are being dragged over it.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
HWND          hwndCnr;    /*  Handle to the container control window. */
PDRAGINFO     pdrgInfo;   /*  Pointer to drag information. */
MRESULT       rc;         /*  Return code. */

rc = _wpDragOver(somSelf, hwndCnr, pdrgInfo);
```

wpDragOver Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpDragOver Parameter - hwndCnr

hwndCnr (HWND) - input
Handle to the container control window.

wpDragOver Parameter - pdrgInfo

pdrgInfo (PDRAGINFO) - input
Pointer to drag information.

wpDragOver Return Value - rc

rc (MRESULT) - returns
Return code.

See DM_DRAGOVER in the *Presentation Manager Programming Reference* for description of the return value.

wpDragOver - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndCnr (HWND) - input
Handle to the container control window.

pdrgInfo (PDRAGINFO) - input
Pointer to drag information.

rc (MRESULT) - returns
Return code.

See DM_DRAGOVER in the *Presentation Manager Programming Reference* for description of the return value.

wpDragOver - Remarks

The wpDragOver method is sent for each DM_DRAGOVER message received by the object. See DM_DRAGOVER in the *Presentation Manager Programming Reference* for more information.

wpDragOver - Usage

This method is generally only called by the system as the folder containing the object processes the DM_DRAGOVER message.

wpDragOver - How to Override

This method should be overridden to determine if the object or objects being dragged can be dropped on this object.

wpDragOver - Related Methods

Related Methods

- wpDrop
- wpFormatDragItem

wpDragOver - Example Code

In this example, objects that are not file system objects are prevented from being dropped on the Browse_O_matic.

```
SOM_Scope MRESULT   SOMLINK Browse_O_Maticwps_wpDragOver(Browse_O_Matic *somSelf,
                HWND hwndCnr,
                PDRAGINFO pdrgInfo)
{
    MRESULT mResult;
    ULONG   ulCount;
    ULONG   ulNumberOfObjects;

    /* Browse_O_MaticData *somThis = Browse_O_MaticGetData(somSelf); */
    Browse_O_MaticMethodDebug("Browse_O_Matic","Browse_O_Maticwps_wpDragOver");

    /*****************************************************************/
    /* Don't call the parent. Initialize mResult to allow the drag over */
    /* to proceed.                                                   */
    /*****************************************************************/
    mResult = MRFROM2SHORT( DOR_DROP, DO_MOVE);

    /*****************************************************************/
    /* Determine the number of objects dragged over the Browse-O-Matic */
    /*****************************************************************/
    ulNumberOfObjects = DrgQueryDragitemCount( pdrgInfo);

    /************************/
    /* Check all the objects */
    /************************/
    for( ulCount=0; ulCount < ulNumberOfObjects &&
                        SHORT1FROMMR( mResult) != DOR_NEVERDROP; ulCount++){

        /***************************************/
        /* It must be a file system type object. */
        /***************************************/
        if( DrgVerifyRMF( DrgQueryDragitemPtr( pdrgInfo, ulCount),
                                                    "DRM_OS2FILE", NULL))
            mResult = MRFROM2SHORT( DOR_DROP, SHORT2FROMMR( mResult));
        else
            mResult = MRFROM2SHORT( DOR_NEVERDROP, SHORT2FROMMR( mResult));
    }

    return( mResult);
}
```

wpDragOver - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpDrop

wpDrop - Syntax

This instance method is called to inform an object that another object has been dropped on it.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
HWND          hwndCnr;   /*  Handle to the container control window. */
PDRAGINFO     pdrgInfo;  /*  Pointer to a DRAGINFO structure. */
PDRAGITEM     pdrgItem;  /*  Pointer to a DRAGITEM structure. */
MRESULT       rc;        /*  Return code. */

rc = _wpDrop(somSelf, hwndCnr, pdrgInfo, pdrgItem);
```

wpDrop Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpDrop Parameter - hwndCnr

hwndCnr (HWND) - input
Handle to the container control window.

wpDrop Parameter - pdrgInfo

pdrgInfo (PDRAGINFO) - input
Pointer to a DRAGINFO structure.

This parameter is used to access all the objects in the drop set, if multiple objects are being dropped.

wpDrop Parameter - pdrgItem

pdrgItem (PDRAGITEM) - input
Pointer to a DRAGITEM structure.

This is the object that is currently being dropped.

wpDrop Return Value - rc

rc (MRESULT) - returns
Return code.

RC_DROP_DROPCOMPLETE
All objects in the drag set have been dropped successfully. wpDrop is not called for the next item in the drop set, and *pdrgInfo* is not freed.

RC_DROP_ERROR
An error occurred while processing the drop operation. The drop operation is terminated, wpDrop is not called for the next item in the drop set, and *pdrgInfo* is freed.

RC_DROP_ITEMCOMPLETE
One object in the drag set, specified by *pdrgItem*, has been dropped successfully. *pdrgInfo* is freed after all objects in the drop set have been dropped.

RC_DROP_RENDERING
The object being dropped has asked for a rendering of the source object. wpDrop is called for the next item in the drop set, *pdrgInfo* is freed after all objects in the drop set have been dropped.

wpDrop - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndCnr (HWND) - input
Handle to the container control window.

pdrgInfo (PDRAGINFO) - input
Pointer to a DRAGINFO structure.

This parameter is used to access all the objects in the drop set, if multiple objects are being dropped.

pdrgItem (PDRAGITEM) - input
Pointer to a DRAGITEM structure.

This is the object that is currently being dropped.

rc (MRESULT) - returns
Return code.

RC_DROP_DROPCOMPLETE
All objects in the drag set have been dropped successfully. wpDrop is not called for the next item in the drop set, and *pdrgInfo* is not freed.

RC_DROP_ERROR
An error occurred while processing the drop operation. The drop operation is terminated, wpDrop is not called for the next item in the drop set, and *pdrgInfo* is freed.

RC_DROP_ITEMCOMPLETE
One object in the drag set, specified by *pdrgItem*, has been dropped successfully. *pdrgInfo* is freed after all objects in the drop set have been dropped.

RC_DROP_RENDERING
The object being dropped has asked for a rendering of the source object. wpDrop is called for the next item in the drop set, *pdrgInfo* is freed after all objects in the drop set have been dropped.

wpDrop - Remarks

The wpDrop method is called when a DM_DROP message is received by the object. See DM_DROP in the *Presentation Manager Programming Reference* for more information.

wpDrop - Usage

This method is generally called only by the system as a folder containing the object processes the DM_DROP message.

wpDrop - How to Override

This method should be overridden to process the action of the dragged object or objects being dropped on it.

wpDrop - Related Methods

Related Methods

- wpDragOver
- wpFormatDragItem

wpDrop - Example Code

In this example, objects that are not file system objects are prevented from being dropped on the Browse_O_matic.

```
SOM_Scope MRESULT   SOMLINK Browse_O_Maticwps_wpDrop(Browse_O_Matic *somSelf,
                HWND hwndCnr,
                PDRAGINFO pdrgInfo,
                PDRAGITEM pdrgItem)
{
    CHAR    pszBuffer[ 255 ];
    MRESULT mResult;

    /* Browse_O_MaticData *somThis = Browse_O_MaticGetData(somSelf); */
    Browse_O_MaticMethodDebug("Browse_O_Matic","Browse_O_Maticwps_wpDrop");

    mResult = MRFROM2SHORT( DOR_DROP, 0);

    /************************************************************/
    /* Don't call the parent. Initialize mResult to allow the drop */
    /* to proceed.                                              */
    /************************************************************/
    if( DOR_NEVERDROP != SHORT1FROMMR( mResult) &&
                                DrgVerifyRMF( pdrgItem, "DRM_OS2FILE", NULL)){

        /* Get the path */
        DrgQueryStrName( pdrgItem->hstrContainerName, sizeof( pszBuffer),
                                                                pszBuffer);

        /* Append the name of the object to the path */
        DrgQueryStrName( pdrgItem->hstrSourceName,
                                      sizeof( pszBuffer) - strlen( pszBuffer),
                                             &pszBuffer[ strlen( pszBuffer)]);
        _wpViewObject( somSelf, NULLHANDLE, OPEN_DEFAULT, (ULONG)pszBuffer);

        mResult = MRFROM2SHORT( DOR_DROP, SHORT2FROMMR( mResult));
    }
    else
        mResult = MRFROMSHORT( DOR_NEVERDROP );

    return( mResult );
}
```

wpDrop - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpDroppedOnObject

wpDroppedOnObject - Syntax

This instance method is called on an object that has just been dragged (a source object) when the target object that it was dropped on does not know what action to perform. This instance method is called only on a source object when that source object has previously responded favorably to a wpDraggedOverObject instance method call.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;          /*  Pointer to the object on which the method is being invoked. */
WPObject     *DroppedOnObject;  /*  Pointer to the object that was dropped on. */
BOOL          rc;               /*  Return code. */

rc = wpDroppedOnObject(somSelf, DroppedOnObject);
```

wpDroppedOnObject Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpDroppedOnObject Parameter - DroppedOnObject

DroppedOnObject (WPObject *) - input
Pointer to the object that was dropped on.

The current target object for the drag operation.

wpDroppedOnObject Return Value - rc

rc (BOOL) - returns
Return code.

TRUE	The drop action was successful.
FALSE	Error occurred.

wpDroppedOnObject - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

DroppedOnObject (WPObject *) - input
Pointer to the object that was dropped on.

The current target object for the drag operation.

rc (BOOL) - returns
Return code.

TRUE	The drop action was successful.
FALSE	Error occurred.

wpDroppedOnObject - Remarks

When you drop on a target object and the source object has said it knows how to handle the drop operation, this method will be invoked on the source object. For example, the program object class supports being dropped on certain other classes of object where the drop action will be taken by the program itself. The drop action provided in WPProgram's override of the wpDroppedOnObject method would be to execute itself as a viewer of the target object. This method is called as a result of a DM_DROP message being sent.

wpDroppedOnObject - Usage

Target objects that do not know how to handle the current drop operation can call this method on the source object that was dropped on it. This method would not normally be called outside the scope of a drag or drop operation.

wpDroppedOnObject - How to Override

Object classes that override the wpDraggedOverObject instance method would normally be expected to also override this method.

wpDroppedOnObject - Related Methods

Related Methods

- wpDragOver
- wpDraggedOverObject
- wpDrop
- wpFormatDragItem

wpDroppedOnObject - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpEditCell

wpEditCell - Syntax

This instance method is called to edit a value in a cell.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPalette     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PCELL          pCell;    /*  Pointer to the CELL structure to be edited. */
HWND           hwndPal;  /*  Palette window handle. */
BOOL           rc;       /*  Success indicator. */

rc = _wpEditCell(somSelf, pCell, hwndPal);
```

wpEditCell Parameter - somSelf

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

wpEditCell Parameter - pCell

pCell (PCELL) - input
Pointer to the CELL structure to be edited.

wpEditCell Parameter - hwndPal

hwndPal (HWND) - input
Palette window handle.

wpEditCell Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpEditCell - Parameters

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

pCell (PCELL) - input
Pointer to the CELL structure to be edited.

hwndPal (HWND) - input
Palette window handle.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpEditCell - Remarks

The default processing for this method by the WPPalette class is to do nothing other than return FALSE.

wpEditCell - Usage

This method is generally called only by the palette window when the user requests to edit the value in the cell. This request is made by selecting a cell and pressing the Enter key or the edit pushbutton or by double-clicking on the cell.

wpEditCell - How to Override

This method should be overridden to handle the edit action.

wpEditCell - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage

wpEjectDisk

wpEjectDisk - Syntax

This method is specific to version 2.1, or higher, of the OS/2 operating system.

This instance method is used to ejection media from drives that support software eject of removable media.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDisk     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG       rc;       /*  Return code from DosDevIOCtl category 8 Function 40. */

rc = _wpEjectDisk(somSelf);
```

wpEjectDisk Parameter - somSelf

somSelf (WPDisk *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDisk.

wpEjectDisk Return Value - rc

rc (ULONG) - returns
Return code from DosDevIOCtl category 8 Function 40.

A return value of NO_ERROR indicates that the drive does not support lock and eject.

wpEjectDisk - Parameters

somSelf (WPDisk *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDisk.

rc (ULONG) - returns

Return code from DosDevIOCtl category 8 Function 40.

A return value of NO_ERROR indicates that the drive does not support lock and eject.

wpEjectDisk - Usage

This method can be called at any time to eject a disk from a drive.

wpEjectDisk - How to Override

This method should not be overridden.

wpEjectDisk - Related Methods

Related Methods

- wpLockDrive
- wpQueryDriveLockStatus

wpEjectDisk - Example Code

```
ULONG   ulStatus = 0;                /* Status returned from _wpEjectDisk */
APIRET  rc       = NO_ERROR;         /* Return code from DosBeep */

 /*
  *  Eject the disk from the drive
  */

   if ((ulStatus=_wpEjectDisk( self )) == 0) {

     somPrintf("Disk ejected.  Waiting for next CD-ROM to be loaded.\n");
     rc = DosBeep( 8000, 500 );          /* Beep to wake up the user */

   } else {

     somPrintf("_wpEjectDisk failed, return code=%u\n", ulStatus);
     return 1;

   } /* endif */
```

wpEjectDisk - Topics

Select an item:
Syntax
Parameters

Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpEndConversation

wpEndConversation - Syntax

This instance method is called to notify the object that the drag or drop operation is complete.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
ULONG         ulItemID;
ULONG         fResult;     /*  Flag indicating whether the operation was performed successfully. */
MRESULT       mresreturn;

mresreturn = _wpEndConversation(somSelf, ulItemID,
              fResult);
```

wpEndConversation Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpEndConversation Parameter - ulItemID

ulItemID (ULONG) - input
This is the *ulItemID* from the DRAGITEM that was contained within the DRAGINFO structure when the object was dropped.

wpEndConversation Parameter - fResult

fResult (ULONG) - input
Flag indicating whether the operation was performed successfully.

See DM_ENDCONVERSATION in the *Presentation Manager Programming Reference* for more information about this parameter.

wpEndConversation Return Value - mresreturn

mresreturn (MRESULT) - returns
See DM_ENDCONVERSATION in the *Presentation Manager Programming Reference* for a description of the return value.

wpEndConversation - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulItemID (ULONG) - input
This is the *ulItemID* from the DRAGITEM that was contained within the DRAGINFO structure when the object was dropped.

fResult (ULONG) - input
Flag indicating whether the operation was performed successfully.

See DM_ENDCONVERSATION in the *Presentation Manager Programming Reference* for more information about this parameter.

mresreturn (MRESULT) - returns
See DM_ENDCONVERSATION in the *Presentation Manager Programming Reference* for a description of the return value.

wpEndConversation - Remarks

The wpEndConversation method is called when the object receives a DM_ENDCONVERSATION message. See DM_ENDCONVERSATION in the *Presentation Manager Programming Reference* for more information.

wpEndConversation - Usage

This method is generally called only by the system as the folder containing the object processed by the DM_ENDCONVERSATION message.

wpEndConversation - How to Override

This method should be overridden if the object needs to do any actions once the drag or drop operation is complete.

wpEndConversation - Related Methods

Related Methods

- wpDragOver
- wpDrop
- wpEndConversation
- wpFormatDragItem
- wpRender
- wpRenderComplete

wpEndConversation - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpFilterMenu

wpFilterMenu - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method filters out options from the object's pop-up menu that do not apply.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject        *somSelf;
FILTERFLAGS     *pFlags;        /*  Pop-up menu flags. */
HWND             hwndCnr;       /*  Handle to container control window. */
BOOL             fMultiSelect;  /*  Multiple menu items flag. */
ULONG            ulMenuType;    /*  The type of menu that is being built. */
ULONG            ulView;        /*  The view that the menu is being invoked on. */
ULONG            ulReserved;    /*  Reserved.  Set to 0. */
BOOL             rc;            /*  New pop-up menu flags for this object. */

rc = _wpFilterMenu(somSelf, pFlags, hwndCnr,
       fMultiSelect, ulMenuType, ulView, ulReserved);
```

wpFilterMenu Parameter - somSelf

somSelf (WPObject *) - input

Points to an object of class WPObject. Pointer to the object on which the method is being invoked.

wpFilterMenu Parameter - pFlags

pFlags (FILTERFLAGS *) - in/out
Pop-up menu flags.

If the flag is set, the corresponding pop-up menu item is available. These flags are *OR*ed together, with the flags already defined by ancestor classes, to specify the standard pop-up menu items that apply to this object.

CTXT_ARRANGE
: Open arrange

CTXT_CLOSE
: Close

CTXT_COPY
: Copy

CTXT_DELETE
: Delete

CTXT_DETAILS
: Open details view

CTXT_FIND
: Open find

CTXT_HELP
: Help

CTXT_ICON
: Open icon view

CTXT_LINK
: Create shadow

CTXT_LOCKUP
: Open lockup

CTXT_MOVE
: Move

CTXT_NEW
: Create another

CTXT_OPEN
: Open

CTXT_PALETTE
: Open palette

CTXT_PRINT
: Print

CTXT_PROGRAM
: Open program

CTXT_REFRESH
: Refresh

CTXT_SELECT
: Open select

CTXT_SHUTDOWN
: Open shutdown

CTXT_SORT
: Open sort

CTXT_SETTINGS
: Open settings

CTXT_SWITCHTO
: Switch to

CTXT_TREE
: Open tree view

CTXT_WINDOW
: Window

wpFilterMenu Parameter - hwndCnr

hwndCnr (HWND) - input

Handle to container control window.

Possible values are described in the following list:

NULLHANDLE	A pop-up request is made on the "whitespace" of an open view.
Other	The handle of the container requesting the pop-up menu.

wpFilterMenu Parameter - fMultiSelect

fMultiSelect (BOOL) - input
Multiple menu items flag.

wpFilterMenu Parameter - ulMenuType

ulMenuType (ULONG) - input
The type of menu that is being built.

May be one of the following values:

MENU_OBJECTPOPUP	Pop-up menu for the object icon.
MENU_OPENVIEWPOPUP	Pop-up menu for an open view.
MENU_FOLDERPULLDOWN	Pull-down menu for a folder.
MENU_EDITPULLDOWN	Pull-down menu for the Edit menu option.
MENU_VIEWPULLDOWN	Pull-down menu for the View menu option.
MENU_SELECTEDPULLDOWN	Pull-down menu for the Selected menu option.
MENU_HELPPULLDOWN	Pull-down menu for the Help menu option.

wpFilterMenu Parameter - ulView

ulView (ULONG) - input
The view that the menu is being invoked on.

OPEN_CONTENTS	Open contents view.
OPEN_TREE	Open tree view.
OPEN_DETAILS	Open details view.
CLOSED_ICON	Closed icon view.

wpFilterMenu Parameter - ulReserved

ulReserved (ULONG) - input
Reserved. Set to 0.

wpFilterMenu Return Value - rc

rc (BOOL) - returns
New pop-up menu flags for this object.

wpFilterMenu - Parameters

somSelf (WPObject *) - input
Points to an object of class WPObject. Pointer to the object on which the method is being invoked.

pFlags (FILTERFLAGS *) - in/out
Pop-up menu flags.

If the flag is set, the corresponding pop-up menu item is available. These flags are *OR*ed together, with the flags already defined by ancestor classes, to specify the standard pop-up menu items that apply to this object.

CTXT_ARRANGE
: Open arrange

CTXT_CLOSE
: Close

CTXT_COPY
: Copy

CTXT_DELETE
: Delete

CTXT_DETAILS
: Open details view

CTXT_FIND
: Open find

CTXT_HELP
: Help

CTXT_ICON
: Open icon view

CTXT_LINK
: Create shadow

CTXT_LOCKUP
: Open lockup

CTXT_MOVE
: Move

CTXT_NEW
: Create another

CTXT_OPEN
: Open

CTXT_PALETTE
: Open palette

CTXT_PRINT
: Print

CTXT_PROGRAM
: Open program

CTXT_REFRESH
: Refresh

CTXT_SELECT
: Open select

CTXT_SHUTDOWN
: Open shutdown

CTXT_SORT
: Open sort

CTXT_SETTINGS
: Open settings

CTXT_SWITCHTO
: Switch to

CTXT_TREE
: Open tree view

CTXT_WINDOW
: Window

hwndCnr (HWND) - input
Handle to container control window.

Possible values are described in the following list:

NULLHANDLE	A pop-up request is made on the "whitespace" of an open view.
Other	The handle of the container requesting the pop-up menu.

fMultiSelect (BOOL) - input
Multiple menu items flag.

ulMenuType (ULONG) - input
The type of menu that is being built.

May be one of the following values:

MENU_OBJECTPOPUP	Pop-up menu for the object icon.
MENU_OPENVIEWPOPUP	Pop-up menu for an open view.
MENU_FOLDERPULLDOWN	Pull-down menu for a folder.
MENU_EDITPULLDOWN	Pull-down menu for the Edit menu option.
MENU_VIEWPULLDOWN	Pull-down menu for the View menu option.
MENU_SELECTEDPULLDOWN	Pull-down menu for the Selected menu option.
MENU_HELPPULLDOWN	Pull-down menu for the Help menu option.

ulView (ULONG) - input
The view that the menu is being invoked on.

OPEN_CONTENTS	Open contents view.
OPEN_TREE	Open tree view.
OPEN_DETAILS	Open details view.
CLOSED_ICON	Closed icon view.

ulReserved (ULONG) - input
Reserved. Set to 0.

rc (BOOL) - returns
New pop-up menu flags for this object.

wpFilterMenu - Remarks

The wpFilterMenu method is a superset of the OS/2 Warp Version 3 wpFilterPopupMenu method. All new or modified object classes that formerly overrode the wpFilterPopupMenu method should override the wpFilterMenu method instead. This gives you more control over what menu items are filtered from the pop-up menu as well as the folder pull-down menus.

wpFilterMenu - Usage

This method is generally called only by the system when a request is made to display the object's pop-up window or pull-down menu.

wpFilterMenu - How to Override

This method should be overridden to remove undesired pop-up or pull-down menu actions that were added by ancestor classes. The parent method should be called prior to any override processing.

wpFilterMenu - Related Methods

Related Methods

- wpFilterPopupMenu
- wpInsertMenuItems
- wpInsertPopupMenuItems
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyMenu
- wpModifyPopupMenu

wpFilterMenu - Example Code

In this example, the Tree view option is removed from the Open popup menu and the Delete option is added.

```
SOM_Scope BOOL SOMLINK TextFolderwps_wpFilterMenu(TextFolder *somSelf,
                       PFILTERFLAGS pFlags,
                       HWND hwndCnr,
                       BOOL32 fMultiSelect,
                       ULONG ulMenuType,
                       ULONG ulView,
                       ULONG ulReserved)

{
    BOOL     bParentResult;

    /* TextFolderData *somThis = TextFolderGetData(somSelf); */
    TextFolderMethodDebug("TextFolder","TextFolderwps_wpFilterMenu");

    bParentResult = parent_wpFilterMenu(somSelf,
                                        pFlags,
                                        hwndCnr,
                                        fMultiSelect,
                                        ulMenuType,
                                        ulView,
                                        ulReserved);

    if (ulMenuType == MENU_OPENVIEWPOPUP)
    {

       /* This is a pop-up menu - remove the Tree view menu option and
        * make sure the Delete option is available
        */
       pFlags->Flags[0] |= CTXT_DELETE;
       pFlags->Flags[0] &= ~CTXT_TREE;
    }

    return bParentResult;
}
```

wpFilterMenu - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpFilterPopupMenu

wpFilterPopupMenu - Syntax

This instance method is called to allow the object to modify its context menu.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
ULONG         ulFlags;       /*  Pop-up menu flags. */
HWND          hwndCnr;       /*  Handle to container control window. */
BOOL          fMultiSelect;  /*  Multiple menu items flag. */
ULONG         rc;            /*  New pop-up menu flags for this object. */

rc = _wpFilterPopupMenu(somSelf, ulFlags,
       hwndCnr, fMultiSelect);
```

wpFilterPopupMenu Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpFilterPopupMenu Parameter - ulFlags

ulFlags (ULONG) - input
Pop-up menu flags.

If the flag is set, the corresponding pop-up menu item is available. These flags are *ored* together, with the flags already defined by ancestor classes, to specify the standard pop-up menu items that apply to this object.

CTXT_ARRANGE
: Open arrange

CTXT_CLOSE
: Close

CTXT_COPY
: Copy

CTXT_DELETE
: Delete

CTXT_DETAILS
: Open details view

CTXT_FIND
: Open find

CTXT_HELP
: Help

CTXT_ICON
: Open icon view

CTXT_LINK
: Create shadow

CTXT_LOCKUP
: Open lockup

CTXT_MOVE
: Move

CTXT_NEW
: Create another

CTXT_OPEN
: Open

CTXT_PALETTE
: Open palette

CTXT_PRINT
: Print

CTXT_PROGRAM
: Open program

CTXT_REFRESH
: Refresh

CTXT_SELECT
: Open select

CTXT_SHUTDOWN
: Open shutdown

CTXT_SORT
: Open sort

CTXT_SETTINGS
: Open settings

CTXT_SWITCHTO
: Switch to

CTXT_TREE
: Open tree view

CTXT_WINDOW
: Window

wpFilterPopupMenu Parameter - hwndCnr

hwndCnr (HWND) - input
: Handle to container control window.

Possible values are described in the following list:

NULLHANDLE
: A pop-up request is made on the "whitespace" of an open view.

other
: The handle of the container requesting the pop-up menu.

wpFilterPopupMenu Parameter - fMultiSelect

fMultiSelect (BOOL) - input
: Multiple menu items flag.

wpFilterPopupMenu Return Value - rc

rc (ULONG) - returns
: New pop-up menu flags for this object.

wpFilterPopupMenu - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulFlags (ULONG) - input
Pop-up menu flags.

If the flag is set, the corresponding pop-up menu item is available. These flags are *ored* together, with the flags already defined by ancestor classes, to specify the standard pop-up menu items that apply to this object.

CTXT_ARRANGE
: Open arrange

CTXT_CLOSE
: Close

CTXT_COPY
: Copy

CTXT_DELETE
: Delete

CTXT_DETAILS
: Open details view

CTXT_FIND
: Open find

CTXT_HELP
: Help

CTXT_ICON
: Open icon view

CTXT_LINK
: Create shadow

CTXT_LOCKUP
: Open lockup

CTXT_MOVE
: Move

CTXT_NEW
: Create another

CTXT_OPEN
: Open

CTXT_PALETTE
: Open palette

CTXT_PRINT
: Print

CTXT_PROGRAM
: Open program

CTXT_REFRESH
: Refresh

CTXT_SELECT
: Open select

CTXT_SHUTDOWN
: Open shutdown

CTXT_SORT
: Open sort

CTXT_SETTINGS
: Open settings

CTXT_SWITCHTO
: Switch to

CTXT_TREE
: Open tree view

CTXT_WINDOW
: Window

hwndCnr (HWND) - input
Handle to container control window.

Possible values are described in the following list:

NULLHANDLE
: A pop-up request is made on the "whitespace" of an open view.

other
: The handle of the container requesting the pop-up menu.

fMultiSelect (BOOL) - input
Multiple menu items flag.

rc (ULONG) - returns
New pop-up menu flags for this object.

wpFilterPopupMenu - Usage

This method is generally called only by the system when a request is made to display the object's pop-up window.

wpFilterPopupMenu - How to Override

This method should be overridden to remove undesired pop-up menu actions that were added by ancestor classes. The parent method should be called prior to any override processing.

wpFilterPopupMenu - Related Methods

Related Methods

- wpFilterPopupMenu
- wpInsertPopupMenuItems
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyPopupMenu

wpFilterPopupMenu - Example Code

In this example, the Tree view option is removed from the Open popup menu.

```
SOM_Scope ULONG   SOMLINK TextFolderwps_wpFilterPopupMenu(TextFolder *somSelf,
                ULONG ulFlags,
                HWND hwndCnr,
                BOOL fMultiSelect)

{
    /* TextFolderData *somThis = TextFolderGetData(somSelf); */
    TextFolderMethodDebug("TextFolder","TextFolderwps_wpFilterPopupMenu");

    ulFlags = parent_wpFilterPopupMenu(somSelf,ulFlags,hwndCnr,fMultiSelect);

    /* Remove the tree view menu option and make sure delete is available */
    return ( ( ulFlags | CTXT_DELETE) & ~CTXT_TREE );
}
```

wpFilterPopupMenu - Topics

Select an item:
Syntax
Parameters

Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpFindMinWindow

wpFindMinWindow - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This method returns the minimized window object that corresponds to the given frame window handle.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPMinWinViewer     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
HWND                hwndFrame;   /*  Frame window handle. */
WPObject           *Object;      /*  Pointer to the object that represents the minimized window. */

Object = _wpFindMinWindow(somSelf, hwndFrame);
```

wpFindMinWindow Parameter - somSelf

somSelf (WPMinWinViewer *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPMinWinViewer.

wpFindMinWindow Parameter - hwndFrame

hwndFrame (HWND) - input
Frame window handle.

wpFindMinWindow Return Value - Object

Object (WPObject *) - returns
Pointer to the object that represents the minimized window.

A return value of NULL indicates that no object exists.

wpFindMinWindow - Parameters

somSelf (WPMinWinViewer *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPMinWinViewer.

hwndFrame (HWND) - input
Frame window handle.

Object (WPObject *) - returns
Pointer to the object that represents the minimized window.

A return value of NULL indicates that no object exists.

wpFindMinWindow - How to Override

This method is generally not overridden.

wpFindMinWindow - Related Methods

Related Methods

- wpQueryMinWindow
- wpSetMinWindow

wpFindMinWindow - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpFindTaskRec

wpFindTaskRec - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

An object is sent this message to request information specific to the task being performed in a task thread.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PTASKREC      rc;       /*  Pointer to the task record information. */

rc = _wpFindTaskRec(somSelf);
```

wpFindTaskRec Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpFindTaskRec Return Value - rc

rc (PTASKREC) - returns
Pointer to the task record information.

wpFindTaskRec - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

rc (PTASKREC) - returns
Pointer to the task record information.

wpFindTaskRec - Remarks

The task records are used by the Workplace Shell to keep track of work being done on another thread (such as moving and copying objects).

wpFindTaskRec - How to Override

This method is not generally overridden.

wpFindTaskRec - Related Methods

Related Methods

- wpSetTaskRec

wpFindTaskRec - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpFindUseItem

wpFindUseItem - Syntax

This instance method is called to retrieve an item type from the object's in-use list.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
ULONG         type;          /*  Specify the usage type of the item to be located. */
PUSEITEM      pCurrentItem;  /*  Pointer to a USEITEM structure. */
PUSEITEM      rc;            /*  Success indicator. */

rc = _wpFindUseItem(somSelf, type, pCurrentItem);
```

wpFindUseItem Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpFindUseItem Parameter - type

type (ULONG) - input
Specify the usage type of the item to be located.

wpFindUseItem Parameter - pCurrentItem

pCurrentItem (PUSEITEM) - input
Pointer to a USEITEM structure.

NULL	Retrieve the first item in the in-use list that has a usage type of *type* .
Other	Retrieve the next item in the in-use list, following the item specified by *pCurrentItem* , that has a usage type of *type* .

wpFindUseItem Return Value - rc

rc (PUSEITEM) - returns
Success indicator.

NULL
No items matching the specified find criteria were found in the in-use list for this object.

Other
A pointer to a USEITEM structure that matches the specified find criteria.

wpFindUseItem - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

type (ULONG) - input
Specify the usage type of the item to be located.

pCurrentItem (PUSEITEM) - input
Pointer to a USEITEM structure.

NULL	Retrieve the first item in the in-use list that has a usage type of *type* .
Other	Retrieve the next item in the in-use list, following the item specified by *pCurrentItem* , that has a usage type of *type* .

rc (PUSEITEM) - returns
Success indicator.

NULL
No items matching the specified find criteria were found in the in-use list for this object.

Other

A pointer to a USEITEM structure that matches the specified find criteria.

wpFindUseItem - Remarks

This method will search the object's in-use list for all items that were added by previous calls to the wpAddToObjUseList method.

wpFindUseItem - Usage

This method should be called to determine how the object is currently being used, for example, which views are currently open and what container window the object is inserted into.

wpFindUseItem - How to Override

This method is generally not overridden.

wpFindUseItem - Related Methods

Related Methods

- wpAddToObjUseList
- wpDeleteFromObjUseList

wpFindUseItem - Example Code

In this example, when the last view of a MYFOLDER object is closed, we zero out its first view instance variable, so that the next time it has a first view opened, we can keep track of that.

```
SOM_Scope BOOL  SOMLINK myfold_wpDeleteFromObjUseList(MYFOLDER *somSelf,
                        PUSEITEM  pUseItem )
{
   MYFOLDERData *somThis   = MYFOLDERGetData(somSelf);
   PUSEITEM      pItem;
   BOOL          rcParentCall;
   MYFOLDERMethodDebug("MYFOLDER","myfold_wpDeleteFromObjUseList");

          /* Call parent's method first */

   rcParentCall = parent_wpDeleteFromObjUseList(somSelf, pUseItem);

   pItem = _wpFindUseItem(somSelf, USAGE_OPENVIEW, NULL);

   if (!pItem) {

      _setFirstViewHandle(somSelf, NULLHANDLE);

   } /* endif */

   return( rcParentCall );
}
```

wpFindUseItem - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpFindViewItem

wpFindViewItem - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method locates either the first or next occurrence of one of the specified views in the in-use list.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
ULONG         flViews;       /*  Flag indicating the views to search for. */
PVIEWITEM     pCurrentItem;  /*  Pointer to the previous view item in the in-use list. */
PVIEWITEM     pNextItem;     /*  Pointer to the view item that was found. */

pNextItem = _wpFindViewItem(somSelf, flViews,
              pCurrentItem);
```

wpFindViewItem Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpFindViewItem Parameter - flViews

flViews (ULONG) - input
Flag indicating the views to search for.

VIEW_CONTENTS
Icon view.
VIEW_SETTINGS
Settings notebook.
VIEW_HELP
Help view.
VIEW_RUNNING
Program running.
VIEW_DETAILS
Details view.
VIEW_TREE
Tree view.
VIEW_ANY
Any view item.

wpFindViewItem Parameter - pCurrentItem

pCurrentItem (PVIEWITEM) - input
Pointer to the previous view item in the in-use list.

A value of NULL indicates that the current view item is the first item in the list.

wpFindViewItem Return Value - pNextItem

pNextItem (PVIEWITEM) - returns
Pointer to the view item that was found.

wpFindViewItem - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

flViews (ULONG) - input
Flag indicating the views to search for.

VIEW_CONTENTS
Icon view.
VIEW_SETTINGS
Settings notebook.
VIEW_HELP
Help view.
VIEW_RUNNING
Program running.
VIEW_DETAILS
Details view.
VIEW_TREE
Tree view.
VIEW_ANY

Any view item.

pCurrentItem (PVIEWITEM) - input
Pointer to the previous view item in the in-use list.

A value of NULL indicates that the current view item is the first item in the list.

pNextItem (PVIEWITEM) - returns
Pointer to the view item that was found.

wpFindViewItem - Usage

This method can be called at any time to find either the first or next view item.

wpFindViewItem - How to Override

This method is generally not overridden.

wpFindViewItem - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Glossary

wpFlushNotifications

wpFlushNotifications - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method forces all pending file system notifications for a folder to be processed immediately.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL          rc;       /*  Success indicator */

rc = _wpFlushNotifications(somSelf);
```

wpFlushNotifications Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpFlushNotifications Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpFlushNotifications - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpFlushNotifications - How to Override

This method should not be overridden.

wpFlushNotifications - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Glossary

wpFormatDragItem

wpFormatDragItem - Syntax

This instance method is called to allow the object to format its drag information when the user starts to drag it.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
PDRAGITEM     pdrgItem;  /*  Address of the drag item. */
BOOL          rc;        /*  Success indicator. */

rc = _wpFormatDragItem(somSelf, pdrgItem);
```

wpFormatDragItem Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpFormatDragItem Parameter - pdrgItem

pdrgItem (PDRAGITEM) - input
Address of the drag item.

wpFormatDragItem Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpFormatDragItem - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pdrgItem (PDRAGITEM) - input
Address of the drag item.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpFormatDragItem - Remarks

This method enables the direct manipulation of this object by initializing the DRAGITEM structure.

wpFormatDragItem - Usage

This method is generally called only by the system when the user first starts to drag the object.

wpFormatDragItem - How to Override

This method is generally overridden by classes that require special processing to initiate a drag or drop operation.

wpFormatDragItem - Related Methods

Related Methods

- wpDragOver
- wpDrop

wpFormatDragItem - Example Code

In this example, a unique rendering mechanism format is specified that will only be understood by instances of class MYFOLDER.

```
SOM_Scope BOOL   SOMLINK myf_wpFormatDragItem(MYFILE *somSelf,
                 PDRAGITEM pdrgItem)
{
/*  MYFILEData *somThis = MYFILEGetData(somSelf); */
    MYFILEMethodDebug("MYFILE","myf_wpFormatDragItem");

    parent_wpFormatDragItem(somSelf,pdrgItem);

    /* We do NOT want to really let the workplace shell render
     * our object, so change the rendering mechanism and format
```

```
     * to be uniquely ours.
     */

    DrgDeleteStrHandle(pdrgItem->hstrRMF);

    pdrgItem->hstrRMF = DrgAddStrHandle("<DRM_OUROWNSPECIAL,DRF_OBJECT>");

    return( TRUE );
}
```

wpFormatDragItem - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpFree

wpFree - Syntax

This instance method is called to destroy the object, including its persistent image, and de-allocate its associated resources.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL          rc;       /*  Success indicator. */

rc = _wpFree(somSelf);
```

wpFree Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpFree Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpFree - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpFree - Remarks

This method destroys the persistent form of the object and then frees the memory that represented that object. If confirmations are on, wpDelete prompts the user before calling wpFree.

After a call to this method, the "self" pointer is no longer valid.

wpFree - Usage

General destruction of an object should be done with the wpDelete method. This method is generally called only by the system.

wpFree - How to Override

This method is generally overridden by storage classes that permanently remove this object and its associated data or any objects that need to do special processing before deletion.

The parent method must be called after the processing that is done by the overriding method.

wpFree - Related Methods

Related Methods

- wpDelete
- wpUnInitData

wpFree - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpFreeCellData

wpFreeCellData - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method frees the data for the specified cell.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPalette     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PCELL          pCell;    /*  Pointer to the cell. */
ULONG          ulIndex;  /*  The number of the cell (0 to xCellCount * yCellCount-1). */
BOOL           rc;       /*  Success indicator */

rc = _wpFreeCellData(somSelf, pCell, ulIndex);
```

wpFreeCellData Parameter - somSelf

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

wpFreeCellData Parameter - pCell

pCell (PCELL) - input
Pointer to the cell.

wpFreeCellData Parameter - ulIndex

ulIndex (ULONG) - input
The number of the cell (0 to xCellCount * yCellCount-1).

wpFreeCellData Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpFreeCellData - Parameters

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

pCell (PCELL) - input
Pointer to the cell.

ulIndex (ULONG) - input
The number of the cell (0 to xCellCount * yCellCount-1).

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpFreeCellData - Usage

This method is called by the wpUninitData method for each cell in a palette that is being made dormant. The wpFreeCellData method is responsible for freeing any storage allocated for the cell.

This method should be called after calling wpSaveCellData and wpRestoreCellData to free the memory used by wpSaveCellData.

wpFreeCellData - How to Override

This method must be overridden in any subclass of WPPalette that allocates storage associated with a palette cell.

This method should be overriden by all subclasses that wish to save and restore part of their cell data. For example, the scheme palette will

save the name of the background bitmap for the scheme.

wpFreeCellData - Related Methods

Related Methods

- wpRestoreCellData
- wpSaveCellData

wpFreeCellData - Example Code

In this example, the Scheme Palette has allocated three strings that are pointed to by its CELL data structure. The wpFreeCellData method frees these strings before allowing WPPalette to free the data structure itself.

```
SOM_Scope BOOL   SOMLINK sch_wpFreeCellData(WPSchemePalette *self,
             PCELL pCell, ULONG ulIndex)
{

   /* Free the strings associated with the cell   */
   FreeMem((PBYTE)(((PSCHEME)pCell)->pszAppName));
   FreeMem((PBYTE)(((PSCHEME)pCell)->SaveFolderBackground.pszImageFile));
   FreeMem((PBYTE)(((PSCHEME)pCell)->SaveDesktopBackground.pszImageFile));

   /* Free the cell information   */
   return (parent_wpFreeCellData(self,pCell,ulIndex));
}
```

wpFreeCellData - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpFreeIconPosData

wpFreeIconPosData - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method frees the .ICONPOS extended attribute (EA) information that is allocated by wpInitIconPosData.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */

wpFreeIconPosData(somSelf);
```

wpFreeIconPosData Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpFreeIconPosData - Return Value

There is no return value for this method.

wpFreeIconPosData - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

There is no return value for this method.

wpFreeIconPosData - Remarks

This method frees memory that was allocated by the wpInitIconPosData method.

wpFreeIconPosData - Usage

You must call the wpFreeIconPosData method after you are finished using the icon position data that was loaded by calling the wpInitIconPosData method.

wpFreeIconPosData - How to Override

This method is not generally overridden.

wpFreeIconPosData - Related Methods

Related Methods

- wpIdentify
- wpInitIconPosData
- wpQueryIconPosition
- wpStoreIconPosData

wpFreeIconPosData - Example Code

This example shows how you can use the wpInitIconPosData, wpIdentify, wpQueryIconPosition, and wpFreeIconPosData methods to find the icon position information for all objects within a folder. In this example, *somSelf* contains the pointer to the folder being examined.

```
BOOL        bSem;
WPObject    *Object;
CHAR        szIdentity[CCHMAXSTRING];
POINTL      ptl;
ULONG       ulIndex;

/* Retrieve the icon position information for all objects within the
 * folder */
_wpInitIconPosData(somSelf);

/* Find the every object within the folder   */
bSem = !_wpRequestObjectMutexSem(somSelf,SEM_INDEFINITE_WAIT);
for (Object = _wpQueryContent(somSelf,NULL,QC_FIRST);
     Object;
     Object = _wpQueryContent(somSelf,Object,QC_NEXT));
{

   /* For each object in the folder, find the position of its icon
    * in an Icon View of the folder  */
   if (_wpIdentify(Object,szIdentity))
   {
      if (_wpQueryIconPosition(somSelf,szIdentity,&ptl,&ulIndex))
      {
         /***** ptl contains the position of the icon for this object *****/
      }
   }
}
if (bSem)
{
   _wpReleaseObjectMutexSem(somSelf);
}

/* Free the icon position data structure   */
_wpFreeIconPosData(somSelf);
```

wpFreeIconPosData - Topics

Select an item:
Syntax

Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpFreeMem

wpFreeMem - Syntax

This instance method is called to de-allocate memory allocated by a call to wpAllocMem.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PBYTE         pByte;    /*  Pointer to the memory to be de-allocated. */
BOOL          rc;       /*  Success indicator. */

rc = _wpFreeMem(somSelf, pByte);
```

wpFreeMem Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpFreeMem Parameter - pByte

pByte (PBYTE) - input
Pointer to the memory to be de-allocated.

wpFreeMem Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpFreeMem - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pByte (PBYTE) - input
Pointer to the memory to be de-allocated.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpFreeMem - Remarks

This method de-allocates memory for an object. wpFreeMem should always be called when the memory allocated by wpAllocMem is no longer needed.

wpFreeMem - Usage

This method should be called when the memory allocated by a call to wpAllocMem is no longer needed.

wpFreeMem - How to Override

This method should be overriden only to provide the de-allocation of memory allocated by an override method of wpAllocMem.

wpFreeMem - Related Methods

Related Methods

- wpAllocMem

wpFreeMem - Example Code

This example clears up memory that was allocated on wpInitData.

```
SOM_Scope void   SOMLINK myf_wpUnInitData(MYFILE *somSelf)
{
    MYFILEData *somThis = MYFILEGetData(somSelf);
    MYFILEMethodDebug("MYFILE","myf_wpUnInitData");

    if (_pszLastWorker)
    {
      _wpFreeMem(somSelf,
                 (PBYTE)_pszLastWorker);  /* Free this area */
    }

    parent_wpUnInitData(somSelf);  /* Call parent method last */

}
```

wpFreeMem - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpGetFattrsFromPsz

wpGetFattrsFromPsz - Syntax

This method is specific to Version 3, or higher, of the OS/2 operating system.

This instance method returns the font attributes from a PresParam string.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFontPalette     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PSZ                pszFont;  /*  PresParam string. */
PFATTRS            pFattrs;  /*  Pointer to the font attribute structure. */

wpGetFattrsFromPsz(somSelf, pszFont, pFattrs);
```

wpGetFattrsFromPsz Parameter - somSelf

somSelf (WPFontPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFontPalette.

wpGetFattrsFromPsz Parameter - pszFont

pszFont (PSZ) - input
PresParam string.

wpGetFattrsFromPsz Parameter - pFattrs

pFattrs (PFATTRS) - output
Pointer to the font attribute structure.

wpGetFattrsFromPsz - Return Value

There is no return value for this method.

wpGetFattrsFromPsz - Parameters

somSelf (WPFontPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFontPalette.

pszFont (PSZ) - input
PresParam string.

pFattrs (PFATTRS) - output
Pointer to the font attribute structure.

There is no return value for this method.

wpGetFattrsFromPsz - How to Override

This method is generally not overridden.

wpGetFattrsFromPsz - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Glossary

wpGetTrueClassName

wpGetTrueClassName - Syntax

This method is specific to Version 3, or higher, of the OS/2 operating system.

This instance method returns the class name for the given object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

SOMClassMgr     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
WPObject        *Object;        /*  The object to get the class name for. */
PSZ              pszClassName;  /*  The class name string. */

pszClassName = _wpGetTrueClassName(somSelf,
                Object);
```

wpGetTrueClassName Parameter - somSelf

somSelf (SOMClassMgr *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class SOMClassMgr.

wpGetTrueClassName Parameter - Object

Object (WPObject *) - input
The object to get the class name for.

wpGetTrueClassName Return Value - pszClassName

pszClassName (PSZ) - returns
The class name string.

wpGetTrueClassName - Parameters

somSelf (SOMClassMgr *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class SOMClassMgr.

Object (WPObject *) - input
The object to get the class name for.

pszClassName (PSZ) - returns
The class name string.

wpGetTrueClassName - Remarks

If a standard Workplace Shell object class has been replaced, the wpGetTrueClassName method will return the name of the class that actually implements the object, while the somGetClassName method will return the name of the replacement class.

For example, if the WPFolder class has been replaced by the MyFolder class, wpGetTrueClassName will return the string "WPFolder", while somGetClassName will return the string "MyFolder".

wpGetTrueClassName - Usage

This method can be called at any time to obtain the name of the object class that implements this object.

This method is usually only for use by base classes when the objects they are storing are being made dormant. It is invoked using the SOMClassMgrObject class.

wpGetTrueClassName - How to Override

This method is not generally overridden.

wpGetTrueClassName - Related Methods

Related Methods

- wpModuleForClass
- wpReplacementIsInEffect

wpGetTrueClassName - Example Code

This example shows how to call the wpGetTrueClassName method to obtain the name of the object class that implements the object pointed to by somSelf.

```
    PSZ         pszClassName;

    pszClassName = _wpGetTrueClassName(SOMClassMgrObject,somSelf);
```

wpGetTrueClassName - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpHide

wpHide - Syntax

This instance method is called to hide or minimize open views of this object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL          rc;       /*  Success indicator. */

rc = _wpHide(somSelf);
```

wpHide Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpHide Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpHide - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpHide - Remarks

This method will turn all windows owned by this object invisible or, if the system setting is set to provide minimized windows instead of hidden windows, wpHide will minimize all windows owned by this object.

wpHide - Usage

This method can be called to hide an object's window.

wpHide - How to Override

This method is not generally overridden.

wpHide - Related Methods

Related Methods

- wpHide
- wpRestore

wpHide - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpHideFldrRunObjs

wpHideFldrRunObjs - Syntax

For work area folders, this instance method is called to allow the folder to hide or minimize its open objects.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL          fHide;    /*  Flag indicating whether to hide or show open objects. */
BOOL          rc;       /*  Success indicator. */

rc = _wpHideFldrRunObjs(somSelf, fHide);
```

wpHideFldrRunObjs Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpHideFldrRunObjs Parameter - fHide

fHide (BOOL) - input
Flag indicating whether to hide or show open objects.

TRUE	Hide all objects opened from this folder.
FALSE	Show all objects opened from this folder.

wpHideFldrRunObjs Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpHideFldrRunObjs - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

fHide (BOOL) - input
Flag indicating whether to hide or show open objects.

TRUE	Hide all objects opened from this folder.
FALSE	Show all objects opened from this folder.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpHideFldrRunObjs - Remarks

Open objects in this folder will either be hidden or minimized, depending on the current system setting. This method is called automatically on folders with the work area flag, FOI_WORKAREA, set when it is hidden or minimized.

wpHideFldrRunObjs - Usage

This method can be called at any time in order to hide or minimize all currently open objects in this folder.

wpHideFldrRunObjs - How to Override

This method is generally not overridden.

wpHideFldrRunObjs - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Glossary

wpHoldJob

wpHoldJob - Syntax

This instance method is called to hold a job object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPJob     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL       rc;       /*  Success indicator. */

rc = _wpHoldJob(somSelf);
```

wpHoldJob Parameter - somSelf

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

wpHoldJob Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpHoldJob - Parameters

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpHoldJob - How to Override

This method is generally not overridden.

wpHoldJob - Related Methods

Related Methods

- wpReleaseJob

wpHoldJob - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpHoldPrinter

wpHoldPrinter - Syntax

This instance method is called to hold a printer object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPrinter     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL           rc;       /*  Success indicator. */

rc = _wpHoldPrinter(somSelf);
```

wpHoldPrinter Parameter - somSelf

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

wpHoldPrinter Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpHoldPrinter - Parameters

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpHoldPrinter - How to Override

This method is generally not overridden.

wpHoldPrinter - Related Methods

Related Methods

- wpReleasePrinter

wpHoldPrinter - Example Code

```
VOID Abort_All_Jobs( SOMAny *self )

{

   CHAR   szTitle[256];           /* Title of printer object */

   strcpy( szTitle, _wpQueryTitle( self ) );

   somPrintf("%s: Deleting all jobs.\n", szTitle);

   _wpSetDefaultPrinter( self );

   _wpHoldPrinter( self );

   _wpDeleteAllJobs( self );

}
```

wpHoldPrinter - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Example Code
Related Methods
Glossary

wpIdentify

wpIdentify - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns a string that is unique within a given container that identifies the specified object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
PSZ           pszIdentity;  /*  A unique string that identifies the specified object. */
BOOL          rc;           /*  Success indicator. */

rc = _wpIdentify(somSelf, pszIdentity);
```

wpIdentify Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpIdentify Parameter - pszIdentity

pszIdentity (PSZ) - output
A unique string that identifies the specified object.

The string is of the form "CCCCIIII" where CCCC is the class name and IIII is a unique string identifier provided by the base class (the real file name in the case of WPFileSystem).

wpIdentify Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpIdentify - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pszIdentity (PSZ) - output
A unique string that identifies the specified object.

The string is of the form "CCCCIIII" where CCCC is the class name and IIII is a unique string identifier provided by the base class (the real file name in the case of WPFileSystem).

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpIdentify - Remarks

The buffer pointed to by pszIdentify must always be at least CCHMAXIDENT characters in length.

wpIdentify - Usage

This method can be called at any time to obtain a unique identifier for an object.

wpIdentify - How to Override

This method can be overridden in any object class that wants to append a unique qualifier to the end of the standard identity string for an object.

This method should only be overridden by storage classes. You should always call this method first, then append extra information to the pszIdentity string. (For example, WPObject puts in the class name string, but its subclasses are responsible for putting in the unique identifier.)

wpIdentify - Related Methods

Related Methods

- wpFreeIconPosData
- wpInitIconPosData
- wpQueryIconPosition
- wpStoreIconPosData

wpIdentify - Example Code

In this example, we obtain the identity string for an object.

```
CHAR        szIdentity[CCHMAXSTRING];

 _wpQueryIdentity(somSelf,szIdentity);
```

wpIdentify - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpInitCellStructs

wpInitCellStructs - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets up the cell dimensions of a palette object based upon the current count of rows and columns. This method also allocates the cell structures if they are not present.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPalette     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL           rc;       /*  Success indicator. */

rc = _wpInitCellStructs(somSelf);
```

wpInitCellStructs Parameter - somSelf

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

wpInitCellStructs Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpInitCellStructs - Parameters

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.

FALSE
Error occurred.

wpInitCellStructs - How to Override

This method is generally not overridden.

wpInitCellStructs - Related Methods

Related Methods

- wpRedrawCell
- wpSetupCell

wpInitCellStructs - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpInitData

wpInitData - Syntax

This instance method is called to allow the object to initialize its instance data.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */

wpInitData(somSelf);
```

wpInitData Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpInitData - Return Value

There is no return value for this method.

wpInitData - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

There is no return value for this method.

wpInitData - Remarks

This routine is called by the system when the object is created or when it is awakened from the dormant state so that it can initialize all of its instance variables to a known state. By default, memory allocated to instance variables is zerofilled. Note that this method is called before the object's state is known, so it is very important that the object does not try to process any other method while processing this method. Should an object need extra initialization that requires it to invoke other methods, this should be done from wpRestoreState.

When the object is first created, wpSetupOnce should be overridden to perform initialization that is required only once.

The parent method must be called before any processing is done by your overriding method.

wpInitData - Usage

This method is generally called only by the system when the object is awakened.

wpInitData - How to Override

Any class that has instance variables should override this method so that those variables are all initially in a known state. It is essential to pass this method onto the parent class before doing your own processing. If this method is overridden, wpUnInitData should also be overridden in order to de-allocate resources that were allocated by the override processing of wpInitData.

wpInitData - Related Methods

Related Methods

- wpSetup
- wpUnInitData

wpInitData - Example Code

This example initializes state variables and allocates any extra memory that might be needed.

```
SOM_Scope void SOMLINK myf_wpInitData(MYFILE *somSelf)
{
    MYFILEData *somThis = MYFILEGetData(somSelf);
    MYFILEMethodDebug("MYFILE","myf_wpInitData");

    parent_wpInitData(somSelf);    /* Have parent method do initialization */

    _pszLastWorker = (PSZ) _wpAllocMem(somSelf,
                                       CCHMAXPATH+1,  /* Number of bytes */
                                       NULL);         /* Return error message */

    if (_pszLastWorker) {

       strcpy(_pszLastWorker, DEFAULT_LASTWORKER);

    } /* endif */

}
```

wpInitData - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpInitIconPosData

wpInitIconPosData - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method reads the .ICONPOS extended attribute (EA) for the folder directory to determine where to position the object icons within the folder.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL          rc;       /*  Success indicator */

rc = _wpInitIconPosData(somSelf);
```

wpInitIconPosData Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpInitIconPosData Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpInitIconPosData - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpInitIconPosData - Remarks

This method allocates memory, and that memory should be freed using the wpFreeIconPosData method.

wpInitIconPosData - Usage

You can call the wpInitIconPosData method at any time to initialize an internal data structure with information about the positions of all icons in an Icon View of a folder.

wpInitIconPosData - How to Override

This method is not generally overridden.

wpInitIconPosData - Related Methods

Related Methods

- wpFreeIconPosData
- wpIdentify
- wpQueryIconPosition
- wpStoreIconPosData

wpInitIconPosData - Example Code

This example shows how you can use the wpInitIconPosData, wpIdentify wpQueryIconPosition , and wpFreeIconPosData methods to find the icon position information for all objects within a folder. In this example, somSelf contains the pointer to the folder being examined.

```
BOOL        bSem;
WPObject    *Object;
CHAR        szIdentity[CCHMAXSTRING];
POINTL      ptl;
ULONG       ulIndex;

/* Retrieve the icon position information for all objects within the
 * folder  */
_wpInitIconPosData(somSelf);

/* Find the every object within the folder   */
bSem = !_wpRequestObjectMutexSem(somSelf,SEM_INDEFINITE_WAIT);
for (Object = _wpQueryContent(somSelf,NULL,QC_FIRST);
     Object;
     Object = _wpQueryContent(somSelf,Object,QC_NEXT));
{

   /* For each object in the folder, find the position of its icon
    * in an Icon View of the folder   */
   if (_wpIdentify(Object,szIdentity))
   {
      if (_wpQueryIconPosition(somSelf,szIdentity,&ptl,&ulIndex))
      {
      /***** ptl contains the position of the icon for this object *****/
      }
   }
}
if (bSem)
{
   _wpReleaseObjectMutexSem(somSelf);
}

/* Free the icon position data structure   */
_wpFreeIconPosData(somSelf);
```

wpInitIconPosData - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpInsertMenuItems

wpInsertMenuItems - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method is called to allow an object to insert items into its pop-up menu.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
HWND          hwndMenu;     /*  Handle to the pop-up menu. */
ULONG         iPosition;    /*  Position at which to start inserting items. */
HMODULE       hmod;         /*  Module handle where MenuID can be found. */
ULONG         MenuID;       /*  ID of menu to put into pop-up menu. */
ULONG         ulSubMenuId;  /*  ID of submenu to put into pop-up menu. */
BOOL          rc;           /*  Success indicator. */

rc = _wpInsertMenuItems(somSelf, hwndMenu,
       iPosition, hmod, MenuID, ulSubMenuId);
```

wpInsertMenuItems Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpInsertMenuItems Parameter - hwndMenu

hwndMenu (HWND) - input
Handle to the pop-up menu.

wpInsertMenuItems Parameter - iPosition

iPosition (ULONG) - input
Position at which to start inserting items.

wpInsertMenuItems Parameter - hmod

hmod (HMODULE) - input
Module handle where *MenuID* can be found.

wpInsertMenuItems Parameter - MenuID

MenuID (ULONG) - input
ID of menu to put into pop-up menu.

For a listing of WPMENUIDs, see the individual object classes.

wpInsertMenuItems Parameter - ulSubMenuId

ulSubMenuId (ULONG) - input
ID of submenu to put into pop-up menu.

wpInsertMenuItems Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpInsertMenuItems - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndMenu (HWND) - input
Handle to the pop-up menu.

iPosition (ULONG) - input
Position at which to start inserting items.

hmod (HMODULE) - input
Module handle where *MenuID* can be found.

MenuID (ULONG) - input
ID of menu to put into pop-up menu.

For a listing of WPMENUIDs, see the individual object classes.

ulSubMenuId (ULONG) - input
ID of submenu to put into pop-up menu.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpInsertMenuItems - Remarks

This instance method will insert all menu items in ulMenuID *MenuID* into the pop-up or pull-down menu.

Menu item IDs in open cascade must match corresponding open view.

Class-specific menu IDs should be above WPMENUID_USER.

The wpInsertMenuItems method is a superset of the OS/2 Warp version 3 wpInsertPopupMenuItems method. All new or modified object classes that formerly called the wpModifyPopupMenu method should override the wpInsertMenuItems method instead. This gives you more control over what menu items are added to or removed from the popup menu as well as the folder pull-down menus.

wpInsertMenuItems - Usage

This method can be called only during the processing of wpModifyMenu

wpInsertMenuItems - How to Override

This method is generally not overridden.

wpInsertMenuItems - Related Methods

Related Methods

- wpAddUserItemsToPopupMenu
- wpFilterMenu
- wpFilterPopupMenu
- wpInsertPopupMenuItems
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyMenu
- wpModifyPopupMenu

wpInsertMenuItems - Example Code

This example adds an extra option to the context menu to close all but the first view.

```
SOM_Scope BOOL   SOMLINK myfold_wpModifyMenu(MYFOLDER *somSelf,
                 HWND hwndMenu,
                 HWND hwndCnr,
                 ULONG iPosition,
                 ULONG ulMenuType,
                 ULONG ulView,
                 ULONG ulReserved)
 {
     HMODULE    hmod = NULLHANDLE;
     zString    zsPathName;

     /* MYFOLDERData *somThis = MYFOLDERGetData(somSelf);  */
     MYFOLDERMethodDebug("MYFOLDER","myfold_wpModifyMenu");

     if (ulMenuType == MENU_OPENVIEWPOPUP)
     {

        /* This is a pop-up menu - add options to the menu   */
        hmod = _clsQueryModuleHandle(_MYFOLDER);
        if (hmod)
        {
           _wpInsertMenuItems( somSelf, hwndMenu, 0, hmod,
                               ID_CLOSEVIEWSMENU, 0);
        }
     }

     return (parent_wpModifyMenu(somSelf,
                                 hwndMenu,
                                 hwndCnr,
                                 iPosition,
                                 ulMenuType,
                                 ulView,
                                 ulReserved));
 }
```

wpInsertMenuItems - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpInsertPopupMenuItems

wpInsertPopupMenuItems - Syntax

This instance method is called to allow an object to insert items into its pop-up menu.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
HWND          hwndMenu;   /*  Handle to the pop-up menu. */
ULONG         iPosition;  /*  Position at which to start inserting items. */
HMODULE       hmod;       /*  Module handle where MenuID can be found. */
ULONG         MenuID;     /*  ID of menu to put into pop-up menu. */
ULONG         SubMenuId;  /*  ID of submenu to put into pop-up menu. */
BOOL          rc;         /*  Success indicator. */

rc = _wpInsertPopupMenuItems(somSelf, hwndMenu,
       iPosition, hmod, MenuID, SubMenuId);
```

wpInsertPopupMenuItems Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpInsertPopupMenuItems Parameter - hwndMenu

hwndMenu (HWND) - input
Handle to the pop-up menu.

wpInsertPopupMenuItems Parameter - iPosition

iPosition (ULONG) - input
Position at which to start inserting items.

wpInsertPopupMenuItems Parameter - hmod

hmod (HMODULE) - input
Module handle where *MenuID* can be found.

wpInsertPopupMenuItems Parameter - MenuID

MenuID (ULONG) - input
ID of menu to put into pop-up menu.

For a listing of WPMENUIDs, see the individual object classes.

wpInsertPopupMenuItems Parameter - SubMenuId

SubMenuId (ULONG) - input
ID of submenu to put into pop-up menu.

wpInsertPopupMenuItems Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpInsertPopupMenuItems - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndMenu (HWND) - input
Handle to the pop-up menu.

iPosition (ULONG) - input
Position at which to start inserting items.

hmod (HMODULE) - input
Module handle where *MenuID* can be found.

MenuID (ULONG) - input
ID of menu to put into pop-up menu.

For a listing of WPMENUIDs, see the individual object classes.

SubMenuId (ULONG) - input
ID of submenu to put into pop-up menu.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpInsertPopupMenuItems - Remarks

This instance method will insert all menu items in *MenuID* into the pop-up menu.

Menu item IDs in open cascade must match corresponding open view.

Class-specific menu IDs should be above WPMENUID_USER.

wpInsertPopupMenuItems - Usage

This method can be called only during the processing of wpModifyPopupMenu.

wpInsertPopupMenuItems - How to Override

This method is generally not overridden.

wpInsertPopupMenuItems - Related Methods

Related Methods

- wpFilterPopupMenu
- wpInsertPopupMenuItems
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyPopupMenu

wpInsertPopupMenuItems - Example Code

This example adds an extra option to the context menu to close all but the first view.

```
SOM_Scope BOOL   SOMLINK myfold_wpModifyPopupMenu(MYFOLDER *somSelf,
                 HWND hwndMenu,
                 HWND hwndCnr,
                 ULONG iPosition)

{
    HMODULE    hmod = NULLHANDLE;
    zString    zsPathName;
```

```
    /* MYFOLDERData *somThis = MYFOLDERGetData(somSelf); */
    MYFOLDERMethodDebug("MYFOLDER","myfold_wpModifyPopupMenu");

    hmod = _clsQueryModuleHandle(_MYFOLDER);
    if (hmod)
    {
       _wpInsertPopupMenuItems( somSelf, hwndMenu, 0, hmod, ID_CLOSEVIEWSMENU, 0);
    }

    return (parent_wpModifyPopupMenu(somSelf,hwndMenu,hwndCnr,iPosition));
}
```

wpInsertPopupMenuItems - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpInsertSettingsPage

wpInsertSettingsPage - Syntax

This instance method is called to insert a page into the object's Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndNotebook;  /*  Handle to the Settings notebook. */
PPAGEINFO     ppageinfo;     /*  Pointer to the notebook page information. */
ULONG         ulPageId;      /*  Page identifier for the inserted page. */

ulPageId = _wpInsertSettingsPage(somSelf,
             hwndNotebook, ppageinfo);
```

wpInsertSettingsPage Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpInsertSettingsPage Parameter - hwndNotebook

hwndNotebook (HWND) - input
Handle to the Settings notebook.

wpInsertSettingsPage Parameter - ppageinfo

ppageinfo (PPAGEINFO) - input
Pointer to the notebook page information.

wpInsertSettingsPage Return Value - ulPageId

ulPageId (ULONG) - returns
Page identifier for the inserted page.

A return value of 0 indicates that an error occurred.

wpInsertSettingsPage - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndNotebook (HWND) - input
Handle to the Settings notebook.

ppageinfo (PPAGEINFO) - input
Pointer to the notebook page information.

ulPageId (ULONG) - returns
Page identifier for the inserted page.

A return value of 0 indicates that an error occurred.

wpInsertSettingsPage - Usage

This method can be called only during the processing of the wpAddSettingsPages method.

wpInsertSettingsPage - How to Override

This method is generally not overridden.

wpInsertSettingsPage - Related Methods

Related Methods

- wpAddSettingsPages

wpInsertSettingsPage - Example Code

In this example, a Settings notebook page is added to let the user modify the Last Worker's name.

```
SOM_Scope BOOL SOMLINK myf_wpAddSettingsPages(MYFILE *somSelf, HWND hwndNotebook)

{
    PAGEINFO pageinfo;

    MYFILEData *somThis = MYFILEGetData(somSelf);
    MYFILEMethodDebug("MYFILE","myf_wpAddSettingsPages");

    parent_wpAddSettingsPages(somSelf, hwndNotebook);

    memset((PCH)&pageinfo,0,sizeof(PAGEINFO));
    pageinfo.cb                 = sizeof(PAGEINFO);
    pageinfo.hwndPage           = NULLHANDLE;
    pageinfo.usPageStyleFlags   = BKA_MAJOR;
    pageinfo.usPageInsertFlags  = BKA_FIRST;
    pageinfo.pfnwp              = LastWorkerDlgProc;
    pageinfo.resid              = hmod;
    pageinfo.dlgid              = IDD_LASTWORKER;
    pageinfo.pszName            = "LastWorker";
    pageinfo.pCreateParams      = somSelf;
    pageinfo.idDefaultHelpPanel = ID_HELP_LASTWORKER;
    pageinfo.pszHelpLibraryName = szHelpLibrary;

    return _wpInsertSettingsPage( somSelf, hwndNotebook, &pageinfo );
}
```

wpInsertSettingsPage - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpIsCurrentDesktop

wpIsCurrentDesktop - Syntax

This instance method allows the desktop to specify whether it is the active Desktop folder on the system.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL           rc;       /*  Active desktop indicator. */

rc = _wpIsCurrentDesktop(somSelf);
```

wpIsCurrentDesktop Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpIsCurrentDesktop Return Value - rc

rc (BOOL) - returns
Active desktop indicator.

TRUE	This object is the active Desktop.
FALSE	This object is not the active Desktop and should behave as a normal folder.

wpIsCurrentDesktop - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

rc (BOOL) - returns
Active desktop indicator.

TRUE	This object is the active Desktop.
FALSE	This object is not the active Desktop and should behave as a normal folder.

wpIsCurrentDesktop - Remarks

The active desktop is set by the system every time the Workplace Shell is initialized, or any time the user profile is reset by a call to PrfReset. Because there can be only one active desktop and objects cannot change their class, desktop objects and descendants must call wpIsCurrentDesktop to determine if it is the current desktop. Desktop folders which are not active take on the behavior of standard folder objects.

For more information about PrfReset, see the *Presentation Manager Programming Reference* .

wpIsCurrentDesktop - Usage

This method is called at the beginning of every overridden method. If the return is false, the override method should call its parent without doing any override processing. If the return is true, override processing can be done.

wpIsCurrentDesktop - How to Override

This method is generally not overridden.

wpIsCurrentDesktop - Example Code

```
TheObject *pObjtoChange;

     /* Make sure we are not manipulating the Desktop! */

if (_wpIsCurrentDesktop(pObjtoChange)) {
     WinMessageBox(HWND_DESKTOP,
                   HWND_DESKTOP,
                   "Changing your Desktop is not supported!",
                   0,
                   MB_CANCEL | MB_WARNING | MB_SYSTEMMODAL );
} else {

      /* manipulate the object here */

} /* endif */
```

wpIsCurrentDesktop - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code

wpIsDeleteable

wpIsDeleteable - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method determines whether the object can be deleted or not.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
BOOL          flDelete;  /*  Delete flag. */

flDelete = _wpIsDeleteable(somSelf);
```

wpIsDeleteable Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpIsDeleteable Return Value - flDelete

flDelete (BOOL) - returns
Delete flag.

Possible values are described in the following list:

TRUE
The object can be deleted by the shredder.

FALSE
The object cannot be deleted by the shredder.

wpIsDeleteable - Parameters

somSelf (WPObject *) - input

Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

flDelete (BOOL) - returns
Delete flag.

Possible values are described in the following list:

TRUE
The object can be deleted by the shredder.

FALSE
The object cannot be deleted by the shredder.

wpIsDeleteable - How to Override

This method is generally not overridden.

wpIsDeleteable - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Glossary

wpIsDetailsColumnVisible

wpIsDetailsColumnVisible - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method determines whether any hidden column data exists for the current object.

Note: This method is used only in details view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         index;    /*  Column index for the current details view. */
BOOL          Visible;  /*  Flag indicating whether the details column is visible. */

Visible = _wpIsDetailsColumnVisible(somSelf,
            index);
```

wpIsDetailsColumnVisible Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpIsDetailsColumnVisible Parameter - index

index (ULONG) - input
Column index for the current details view.

wpIsDetailsColumnVisible Return Value - Visible

Visible (BOOL) - returns
Flag indicating whether the details column is visible.

Possible values are described in the following list:

TRUE
Details column is visible.
FALSE
Details column is hidden.

wpIsDetailsColumnVisible - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

index (ULONG) - input
Column index for the current details view.

Visible (BOOL) - returns
Flag indicating whether the details column is visible.

Possible values are described in the following list:

TRUE
Details column is visible.
FALSE
Details column is hidden.

wpIsDetailsColumnVisible - How to Override

This method is generally not overridden.

wpIsDetailsColumnVisible - Related Methods

Related Methods

- wpSetDetailsColumnVisibility

wpIsDetailsColumnVisible - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpIsDiskSwapped

wpIsDiskSwapped - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method checks if the removable media on which the file system object resides has been swapped.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
BOOL              fSwapped;  /*  Swap flag. */

fSwapped = _wpIsDiskSwapped(somSelf);
```

wpIsDiskSwapped Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpIsDiskSwapped Return Value - fSwapped

fSwapped (BOOL) - returns
Swap flag.

TRUE	Removable media has been swapped.
FALSE	Removable media has not been swapped.

wpIsDiskSwapped - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

fSwapped (BOOL) - returns
Swap flag.

TRUE	Removable media has been swapped.
FALSE	Removable media has not been swapped.

wpIsDiskSwapped - Usage

This method can be called at any time.

wpIsDiskSwapped - How to Override

This method is generally not overridden.

wpIsDiskSwapped - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Glossary

wpIsLocked

wpIsLocked - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method indicates whether the object can be made dormant.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL          rc;       /*  A flag indicating whether the object can be made dormant. */

rc = _wpIsLocked(somSelf);
```

wpIsLocked Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpIsLocked Return Value - rc

rc (BOOL) - returns
A flag indicating whether the object can be made dormant.

Possible values are described in the following list:

TRUE
The object is locked and cannot be made dormant.

FALSE
The object is not locked and can be made dormant.

wpIsLocked - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

rc (BOOL) - returns
A flag indicating whether the object can be made dormant.

Possible values are described in the following list:

TRUE	The object is locked and cannot be made dormant.
FALSE	The object is not locked and can be made dormant.

wpIsLocked - Usage

This method can be called at any time.

wpIsLocked - How to Override

This method is generally not overridden.

wpIsLocked - Related Methods

Related Methods

- wpLockObject

wpIsLocked - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpIsObjectInitialized

wpIsObjectInitialized - Syntax

This instance method is called to determine if new object initialization has completed.

```
#define INCL_WINWORKPLACE
#include <os2.h>
```

```
WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL          rc;       /*  Initialization indicator. */

rc = _wpIsObjectInitialized(somSelf);
```

wpIsObjectInitialized Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpIsObjectInitialized Return Value - rc

rc (BOOL) - returns
Initialization indicator.

TRUE
Object is initialized.
FALSE
Object is not initialized.

wpIsObjectInitialized - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

rc (BOOL) - returns
Initialization indicator.

TRUE
Object is initialized.
FALSE
Object is not initialized.

wpIsObjectInitialized - Remarks

This method is useful in other methods whose processing may differ depending on whether the object is in the process of being created or has completed creation. Setting the object's state to initialized is one of the final steps of object creation during move, copy, and create operations.

Certain methods, such as wpSaveDeferred and wpSaveImmediate, are only valid on initialized objects.

wpIsObjectInitialized can be used within wpSetup to determine if wpSetup is being called during object creation (wpIsObjectInitialized returns FALSE) or after the object has been created, for example from WinSetObjectData (wpIsObjectInitialized returns TRUE).

wpIsObjectInitialized - How to Override

This method should not be overridden.

wpIsObjectInitialized - Related Methods

Related Methods

- wpObjectReady
- wpSetup

wpIsObjectInitialized - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpIsSortAttribAvailable

wpIsSortAttribAvailable - Syntax

This method is specific to Version 3, or higher, of the OS/2 operating system.

This instance method determines whether sort attributes are available for the current folder object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
ULONG         index;      /*  Column index for the current details view. */
BOOL          Available;  /*  Flag indicating whether sort attributes are visible. */

Available = _wpIsSortAttribAvailable(somSelf,
              index);
```

wpIsSortAttribAvailable Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpIsSortAttribAvailable Parameter - index

index (ULONG) - input
Column index for the current details view.

wpIsSortAttribAvailable Return Value - Available

Available (BOOL) - returns
Flag indicating whether sort attributes are visible.

Possible values are described in the following list:

TRUE
Sort attributes are visible.
FALSE
Sort attributes are hidden.

wpIsSortAttribAvailable - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

index (ULONG) - input
Column index for the current details view.

Available (BOOL) - returns
Flag indicating whether sort attributes are visible.

Possible values are described in the following list:

TRUE
Sort attributes are visible.
FALSE
Sort attributes are hidden.

wpIsSortAttribAvailable - How to Override

This method is generally not overridden.

wpIsSortAttribAvailable - Related Methods

Related Methods

- wpSetSortAttribAvailable

wpIsSortAttribAvailable - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpJobAdded

wpJobAdded - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method notifies the printer object that a job has been added to the print queue.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPrinter     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG          ulJobId;  /*  ID of the job added to the print queue. */
BOOL           rc;       /*  Success indicator. */

rc = _wpJobAdded(somSelf, ulJobId);
```

wpJobAdded Parameter - somSelf

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

wpJobAdded Parameter - ulJobId

ulJobId (ULONG) - input
ID of the job added to the print queue.

wpJobAdded Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpJobAdded - Parameters

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

ulJobId (ULONG) - input
ID of the job added to the print queue.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpJobAdded - Usage

This method can be called at any time.

wpJobAdded - How to Override

This method can be overridden, but the parent must be called for proper functioning of the printer object.

wpJobAdded - Related Methods

Related Methods

- wpJobChanged
- wpJobDeleted

wpJobAdded - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpJobChanged

wpJobChanged - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method notifies the printer object that a job in the print queue has been changed in some way.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPrinter     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG          ulJobId;  /*  ID of the job that was changed. */
ULONG          ulField;  /*  The new status of the job. */
BOOL           rc;       /*  Success indicator. */

rc = _wpJobChanged(somSelf, ulJobId, ulField);
```

wpJobChanged Parameter - somSelf

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

wpJobChanged Parameter - ulJobId

ulJobId (ULONG) - input
ID of the job that was changed.

wpJobChanged Parameter - ulField

ulField (ULONG) - input
The new status of the job.

May be one of the following values:

PO_JOB_HELD
The job is being held.
PO_JOB_RELEASE
The job has been released.

wpJobChanged Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpJobChanged - Parameters

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

ulJobId (ULONG) - input
ID of the job that was changed.

ulField (ULONG) - input
The new status of the job.

May be one of the following values:

PO_JOB_HELD
The job is being held.
PO_JOB_RELEASE
The job has been released.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpJobChanged - Usage

This method can be called at any time.

wpJobChanged - How to Override

This method can be overridden, but the parent must be called for proper functioning of the printer object.

wpJobChanged - Related Methods

Related Methods

- wpJobAdded
- wpJobDeleted

wpJobChanged - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpJobDeleted

wpJobDeleted - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method notifies the printer object that a job has been deleted from the print queue.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPrinter     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
```

```
ULONG           ulJobId;  /*  ID of the job deleted from the print queue. */
BOOL            rc;       /*  Success indicator. */

rc = _wpJobDeleted(somSelf, ulJobId);
```

wpJobDeleted Parameter - somSelf

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

wpJobDeleted Parameter - ulJobId

ulJobId (ULONG) - input
ID of the job deleted from the print queue.

wpJobDeleted Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpJobDeleted - Parameters

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

ulJobId (ULONG) - input
ID of the job deleted from the print queue.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpJobDeleted - Usage

This method can be called at any time.

wpJobDeleted - How to Override

This method can be overridden, but the parent must be called for proper functioning of the printer object.

wpJobDeleted - Related Methods

Related Methods

- wpJobAdded
- wpJobChanged

wpJobDeleted - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpLockDrive

wpLockDrive - Syntax

This method is specific to version 2.1, or higher, of the OS/2 operating system.

This instance method is used to lock or unlock removable media in drives that support this feature.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDisk     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
BOOL        fLock;         /*  Lock flag. */
ULONG       ulReturnCall;  /*  Return code from DosDevIOCtl category 8 Function 40. */
```

```
ulReturnCall = _wpLockDrive(somSelf, fLock);
```

wpLockDrive Parameter - somSelf

somSelf (WPDisk *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDisk.

wpLockDrive Parameter - fLock

fLock (BOOL) - input
Lock flag.

Possible values are described in the following list:

TRUE	Increment the lock count.
FALSE	Decrement the lock count.

wpLockDrive Return Value - ulReturnCall

ulReturnCall (ULONG) - returns
Return code from DosDevIOCtl category 8 Function 40.

A return value of NO_ERROR indicates that the drive does not support lock and eject.

wpLockDrive - Parameters

somSelf (WPDisk *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDisk.

fLock (BOOL) - input
Lock flag.

Possible values are described in the following list:

TRUE	Increment the lock count.
FALSE	Decrement the lock count.

ulReturnCall (ULONG) - returns
Return code from DosDevIOCtl category 8 Function 40.

A return value of NO_ERROR indicates that the drive does not support lock and eject.

wpLockDrive - Usage

This method can be called at any time to lock or unlock a drive.

wpLockDrive - How to Override

This method should not be overridden.

wpLockDrive - Related Methods

Related Methods

- wpEjectDisk
- wpQueryDriveLockStatus

wpLockDrive - Example Code

```
ULONG   ulStatus    = 0;             /* Status returned from _wpLockDrive */
ULONG   ulLockCount = 0;             /* Number of times drive is locked   */

           /* Lock the target drive */

   ulStatus = _wpLockDrive( self, TRUE );

   if (ulStatus & DRIVE_LOCK_NO_SUPPORT) {
     somPrintf("Drive does not support lock/unlock.\n");
   } else {

          /* Query the target drive */

     rc = _wpQueryDriveLockStatus( self                /* Self pointer */
                                 , &ulStatus           /* Lock status  */
                                 , &ulLockCount );     /* Lock count   */

     if (rc == NO_ERROR) {

       somPrintf("_wpQueryDriveLockStatus reports:\n");

       if (ulStatus & DRIVE_LOCKED) {
         somLPrintf(1, "Drive is locked.\n");
         somLPrintf(1, "Lock count = %u\n", ulLockCount);
       }
     }

          /* Unlock the target drive */

     ulStatus = _wpLockDrive( self, FALSE );

   } /* endif */
```

wpLockDrive - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpLockObject

wpLockObject - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method locks an object by incrementing the lock count of the receiving object so that it cannot be made dormant.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */

wpLockObject(somSelf);
```

wpLockObject Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpLockObject - Return Value

There is no return value for this method.

wpLockObject - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

There is no return value for this method.

wpLockObject - Usage

This method can be called at any time to lock the object.

wpLockObject - How to Override

This method is generally not overridden.

wpLockObject - Related Methods

Related Methods

- wpIsLocked

wpLockObject - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpMenuItemHelpSelected

wpMenuItemHelpSelected - Syntax

This instance method is called to allow the object to display the requested help panel.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         MenuId;   /*  Object's menu ID. */
BOOL          rc;       /*  Success indicator. */

rc = _wpMenuItemHelpSelected(somSelf, MenuId);
```

wpMenuItemHelpSelected Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpMenuItemHelpSelected Parameter - MenuId

MenuId (ULONG) - input
Object's menu ID.

For a listing of WPMENUIDs, see the individual object classes.

wpMenuItemHelpSelected Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpMenuItemHelpSelected - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

MenuId (ULONG) - input
Object's menu ID.

For a listing of WPMENUIDs, see the individual object classes.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpMenuItemHelpSelected - Remarks

The default WPObject class doesn't process this method at all other than to return false.

wpMenuItemHelpSelected - Usage

This method is generally called only by the system when help on a pop-up menu item is requested.

wpMenuItemHelpSelected - How to Override

This method should be overridden in order to display an appropriate help panel for a user-defined menu item. This is typically accomplished by issuing a call to the wpDisplayHelp method.

wpMenuItemHelpSelected - Related Methods

Related Methods

- wpDisplayHelp
- wpMenuItemHelpSelected
- wpQueryDefaultHelp
- wpSetDefaultHelp
- wpclsQueryDefaultHelp
- wpFilterPopupMenu
- wpInsertPopupMenuItems
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyPopupMenu

wpMenuItemHelpSelected - Example Code

This example overrides wpMenuItemHelpSelected and provides help for the Hex and Text view popup menu selections.

```
SOM_Scope BOOL   SOMLINK Browse_O_Maticwps_wpMenuItemHelpSelected(Browse_O_Matic *somSelf,
                 ULONG MenuId)
{
    /* Browse_O_MaticData *somThis = Browse_O_MaticGetData(somSelf); */
    Browse_O_MaticMethodDebug("Browse_O_Matic","Browse_O_Maticwps_wpMenuItemHelpSelected");

    switch( MenuId){

      case ID_OPEN_TEXT:
         return( _wpDisplayHelp( somSelf, PANEL_HELP_OPEN_TEXT, "pmbrowse.hlp"));

      case ID_OPEN_HEX:
```

```
         return( _wpDisplayHelp( somSelf, PANEL_HELP_OPEN_HEX, "pmbrowse.hlp"));

      default:
         break;
    }

    return( FALSE);
}
```

wpMenuItemHelpSelected - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpMenuItemSelected

wpMenuItemSelected - Syntax

This instance method is called to allow an object to process a pop-up menu selection.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
HWND          hwndFrame;  /*  Handle to the frame window. */
ULONG         ulMenuId;   /*  ID of the selected pop-up menu. */
BOOL          rc;         /*  Success indicator. */

rc = _wpMenuItemSelected(somSelf, hwndFrame,
       ulMenuId);
```

wpMenuItemSelected Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpMenuItemSelected Parameter - hwndFrame

hwndFrame (HWND) - input
Handle to the frame window.

wpMenuItemSelected Parameter - ulMenuId

ulMenuId (ULONG) - input
ID of the selected pop-up menu.

For a listing of WPMENUIDs, see the individual object classes.

wpMenuItemSelected Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpMenuItemSelected - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndFrame (HWND) - input
Handle to the frame window.

ulMenuId (ULONG) - input
ID of the selected pop-up menu.

For a listing of WPMENUIDs, see the individual object classes.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpMenuItemSelected - Remarks

Class-specific menu IDs should be above WPMENUID_USER.

wpMenuItemSelected - Usage

This method is generally called only by the system when a pop-up menu item is selected.

wpMenuItemSelected - How to Override

This method should be overridden to process class-specific menu-item actions or to modify the behavior of a menu-item action provided by an ancestor class.

wpMenuItemSelected - Related Methods

Related Methods

- wpFilterPopupMenu
- wpInsertPopupMenuItems
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyPopupMenu

wpMenuItemSelected - Example Code

This example opens Hex and Text views.

```
SOM_Scope BOOL   SOMLINK Browse_O_Maticwps_wpMenuItemSelected(Browse_O_Matic *somSelf,
                 HWND hwndFrame,
                 ULONG ulMenuId)
{
    /* Browse_O_MaticData *somThis = Browse_O_MaticGetData(somSelf); */
    Browse_O_MaticMethodDebug("Browse_O_Matic","Browse_O_Maticwps_wpMenuItemSelected");

    switch( ulMenuId ){

       case ID_OPEN_TEXT:
          _wpViewObject( somSelf, NULLHANDLE, ulMenuId, 0L );
          break;

       case ID_OPEN_HEX:
          _wpViewObject( somSelf, NULLHANDLE, ulMenuId, 0L );
          break;

       case ID_OPEN_PRODUCTINFO:
       case ID_PRODUCTINFOMENU:
          _wpViewObject( somSelf, NULLHANDLE, ulMenuId, 0L );

       default:
          return( parent_wpMenuItemSelected( somSelf, hwndFrame, ulMenuId) );
    }

    return( TRUE );
}
```

wpMenuItemSelected - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpModifyFldrFlags

wpModifyFldrFlags - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method either sets or removes folder status flags from a given folder.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
ULONG         ulFlags;     /*  Status flags to be set or cleared. */
ULONG         ulFlagMask;  /*  Folder flags in ulFlags to be set or cleared. */
BOOL          rc;          /*  Success indicator. */

rc = _wpModifyFldrFlags(somSelf, ulFlags,
       ulFlagMask);
```

wpModifyFldrFlags Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpModifyFldrFlags Parameter - ulFlags

ulFlags (ULONG) - input
Status flags to be set or cleared.

Possible values are described in the following list: Values can be combined to modify multiple flags in a single call.

FOI_POPULATEWITHALL

FOI_POPULATEWITHFOLDERS

FOI_WORKAREA

FOI_CHANGEFONT

FOI_WAMINIMIZE

FOI_WASTARTONRESTORE

FOI_NOREFRESHVIEWS

FOI_ASYNCREFRESHONOPEN

FOI_TREEPOPULATE

FOI_POPULATEINPROGRESS

FOI_REFRESHINPROGRESS

FOI_FIRSTPOPULATE

FOI_WAMCRINPROGRESS

FOI_CNRBKGNDOLDFORMAT

FOI_CHANGEICONBGNDCOLOR

FOI_CHANGEICONTEXTCOLOR

FOI_DELETEINPROGRESS

FOI_CHANGESHADOWTEXTCOLOR

wpModifyFldrFlags Parameter - ulFlagMask

ulFlagMask (ULONG) - input
Folder flags in *ulFlags* to be set or cleared.

Values specified correspond to the flag values specified in *ulFlags* . If the value in *ulFlagMask* is specified (the bit is 1), the corresponding folder flag is set. If the value in *ulFlagMask* is not specified (the bit is 0), the corresponding folder flag is cleared.

wpModifyFldrFlags Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpModifyFldrFlags - Parameters

somSelf (WPFolder *) - input

Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulFlags (ULONG) - input
Status flags to be set or cleared.

Possible values are described in the following list: Values can be combined to modify multiple flags in a single call.

FOI_POPULATEWITHALL

FOI_POPULATEWITHFOLDERS

FOI_WORKAREA

FOI_CHANGEFONT

FOI_WAMINIMIZE

FOI_WASTARTONRESTORE

FOI_NOREFRESHVIEWS

FOI_ASYNCREFRESHONOPEN

FOI_TREEPOPULATE

FOI_POPULATEINPROGRESS

FOI_REFRESHINPROGRESS

FOI_FIRSTPOPULATE

FOI_WAMCRINPROGRESS

FOI_CNRBKGNDOLDFORMAT

FOI_CHANGEICONBGNDCOLOR

FOI_CHANGEICONTEXTCOLOR

FOI_DELETEINPROGRESS

FOI_CHANGESHADOWTEXTCOLOR

ulFlagMask (ULONG) - input
Folder flags in *ulFlags* to be set or cleared.

Values specified correspond to the flag values specified in *ulFlags*. If the value in *ulFlagMask* is specified (the bit is 1), the corresponding folder flag is set. If the value in *ulFlagMask* is not specified (the bit is 0), the corresponding folder flag is cleared.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpModifyFldrFlags - Remarks

This method sets or clears the corresponding folder flags as an atomic operation. This method can be used instead of wpQueryFldrFlags and wpSetFldrFlags.

wpModifyFldrFlags - Usage

This instance method can be called at any time to change a folder's status.

wpModifyFldrFlags - How to Override

This method is generally not overridden.

wpModifyFldrFlags - Related Methods

Related Methods

- wpQueryFldrFlags
- wpSetFldrFlags

wpModifyFldrFlags - Example Code

The following code marks the folder as a workarea.

```
_wpModifyFldrFlags (self, FOI_WORKAREA ,
                          FOI_WORKAREA);
```

wpModifyFldrFlags - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpModifyMenu

wpModifyMenu - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method is called to allow an object to add additional items to its pop-up menu.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
HWND          hwndMenu;    /*  Menu handle. */
HWND          hwndCnr;     /*  Handle to container control window. */
ULONG         iPosition;   /*  Position to insert menu items. */
ULONG         ulMenuType;  /*  The type of menu that is being built. */
ULONG         ulView;      /*  The view that the menu is being invoked on. */
ULONG         ulReserved;  /*  Reserved.  Set to 0. */
BOOL          rc;          /*  Success indicator. */

rc = _wpModifyMenu(somSelf, hwndMenu, hwndCnr,
       iPosition, ulMenuType, ulView, ulReserved);
```

wpModifyMenu Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpModifyMenu Parameter - hwndMenu

hwndMenu (HWND) - input
Menu handle.

wpModifyMenu Parameter - hwndCnr

hwndCnr (HWND) - input
Handle to container control window.

wpModifyMenu Parameter - iPosition

iPosition (ULONG) - input
Position to insert menu items.

wpModifyMenu Parameter - ulMenuType

ulMenuType (ULONG) - input
The type of menu that is being built.

Can be any one of the following values:

MENU_OBJECTPOPUP	Pop-up menu for the object icon.
MENU_OPENVIEWPOPUP	Pop-up menu for an open view.
MENU_FOLDERPULLDOWN	Pull-down menu for a folder.
MENU_EDITPULLDOWN	Pull-down menu for the Edit menu option.
MENU_VIEWPULLDOWN	Pull-down menu for the View menu option.
MENU_SELECTEDPULLDOWN	Pull-down menu for the Selected menu option.
MENU_HELPPULLDOWN	Pull-down menu for the Help menu option.

wpModifyMenu Parameter - ulView

ulView (ULONG) - input
The view that the menu is being invoked on.

OPEN_CONTENTS	Open contents view.
OPEN_TREE	Open tree view.
OPEN_DETAILS	Open details view.
CLOSED_ICON	Closed icon view.

wpModifyMenu Parameter - ulReserved

ulReserved (ULONG) - input
Reserved. Set to 0.

wpModifyMenu Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpModifyMenu - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndMenu (HWND) - input

Menu handle.

hwndCnr (HWND) - input
Handle to container control window.

iPosition (ULONG) - input
Position to insert menu items.

ulMenuType (ULONG) - input
The type of menu that is being built.

Can be any one of the following values:

MENU_OBJECTPOPUP	Pop-up menu for the object icon.
MENU_OPENVIEWPOPUP	Pop-up menu for an open view.
MENU_FOLDERPULLDOWN	Pull-down menu for a folder.
MENU_EDITPULLDOWN	Pull-down menu for the Edit menu option.
MENU_VIEWPULLDOWN	Pull-down menu for the View menu option.
MENU_SELECTEDPULLDOWN	Pull-down menu for the Selected menu option.
MENU_HELPPULLDOWN	Pull-down menu for the Help menu option.

ulView (ULONG) - input
The view that the menu is being invoked on.

OPEN_CONTENTS	Open contents view.
OPEN_TREE	Open tree view.
OPEN_DETAILS	Open details view.
CLOSED_ICON	Closed icon view.

ulReserved (ULONG) - input
Reserved. Set to 0.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpModifyMenu - Remarks

Class-specific menu IDs should be above WPMENUID_USER. This method should be called only if the current pop-up menu applies to objects of the same class.

wpModifyMenu - Usage

This method is generally called by the system when a request to display the object's pop-up or pull-down menu is made. This method is called following a call to the wpFilterMenu method.

wpModifyMenu - How to Override

This method should be overridden in order to add class-specific actions to the object's pop-up or pull-down menu. Descendant classes can remove these actions by processing the wpFilterMenu method.

wpModifyMenu - Related Methods

Related Methods

- wpAddUserItemsToPopupMenu
- wpFilterMenu
- wpFilterPopupMenu
- wpInsertMenuItems
- wpInsertPopupMenuItems
- wpMenuItemHelpSelected
- wpMenuItemSelected
- wpModifyPopupMenu

wpModifyMenu - Example Code

This example adds an extra option to the context menu to close all but the first view.

```
SOM_Scope BOOL   SOMLINK myfold_wpModifyMenu(MYFOLDER *somSelf,
                 HWND hwndMenu,
                 HWND hwndCnr,
                 ULONG iPosition,
                 ULONG ulMenuType,
                 ULONG ulView,
                 ULONG ulReserved)
 {
     HMODULE    hmod = NULLHANDLE;
     zString    zsPathName;

     /* MYFOLDERData *somThis = MYFOLDERGetData(somSelf); */
     MYFOLDERMethodDebug("MYFOLDER","myfold_wpModifyMenu");

     if (ulMenuType == MENU_OPENVIEWPOPUP)
     {

        /* This is a pop-up menu - add options to the menu   */
        hmod = _clsQueryModuleHandle(_MYFOLDER);
        if (hmod)
        {
           _wpInsertMenuItems( somSelf, hwndMenu, 0, hmod, ID_CLOSEVIEWSMENU, 0);
        }
     }

     return (parent_wpModifyMenu(somSelf,
                                 hwndMenu,
                                 hwndCnr,
                                 iPosition,
                                 ulMenuType,
                                 ulView,
                                 ulReserved));
 }
```

wpModifyMenu - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpModifyPopupMenu

wpModifyPopupMenu - Syntax

This instance method is called to allow an object to add additional items to its pop-up menu.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
HWND          hwndMenu;   /*  Menu handle. */
HWND          hwndCnr;    /*  Handle to container control window. */
ULONG         iPosition;  /*  Position to insert menu items. */
BOOL          rc;         /*  Success indicator. */

rc = _wpModifyPopupMenu(somSelf, hwndMenu,
       hwndCnr, iPosition);
```

wpModifyPopupMenu Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpModifyPopupMenu Parameter - hwndMenu

hwndMenu (HWND) - input
Menu handle.

wpModifyPopupMenu Parameter - hwndCnr

hwndCnr (HWND) - input
Handle to container control window.

wpModifyPopupMenu Parameter - iPosition

iPosition (ULONG) - input
Position to insert menu items.

wpModifyPopupMenu Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpModifyPopupMenu - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndMenu (HWND) - input
Menu handle.

hwndCnr (HWND) - input
Handle to container control window.

iPosition (ULONG) - input
Position to insert menu items.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpModifyPopupMenu - Remarks

Class-specific menu IDs should be above WPMENUID_USER. This method should be called only if the current pop-up menu applies to objects of the same class.

wpModifyPopupMenu - Usage

This method is generally called by the system when a request to display the object's pop-up menu is made. This method is called following a call to the wpFilterPopupMenu method.

wpModifyPopupMenu - How to Override

This method should be overridden in order to add class-specific actions to the object's pop-up menu. Descendant classes can remove these actions by processing the wpFilterPopupMenu method.

wpModifyPopupMenu - Related Methods

Related Methods

wpModifyPopupMenu - Example Code

This example adds an extra option to the context menu to close all but the first view.

```
SOM_Scope BOOL   SOMLINK myfold_wpModifyPopupMenu(MYFOLDER *somSelf,
                 HWND hwndMenu,
                 HWND hwndCnr,
                 ULONG iPosition)

{
    HMODULE    hmod = NULLHANDLE;
    zString    zsPathName;

    /* MYFOLDERData *somThis = MYFOLDERGetData(somSelf); */
    MYFOLDERMethodDebug("MYFOLDER","myfold_wpModifyPopupMenu");

    hmod = _clsQueryModuleHandle(_MYFOLDER);
    if (hmod)
    {
       _wpInsertPopupMenuItems( somSelf, hwndMenu, 0, hmod, ID_CLOSEVIEWSMENU, 0);
    }

    return (parent_wpModifyPopupMenu(somSelf,hwndMenu,hwndCnr,iPosition));
}
```

wpModifyPopupMenu - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpModifyStyle

wpModifyStyle - Syntax

This method is specific to Version 3, or higher, of the OS/2 operating system.

This instance method modifies the current object style for the given object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
ULONG         ulStyleFlags;  /*  Flag indicating the object's styles to be affected. */
ULONG         ulStyleMask;   /*  Style flags in ulStyleFlags to be set or cleared. */
BOOL          rc;            /*  Success indicator. */

rc = _wpModifyStyle(somSelf, ulStyleFlags,
       ulStyleMask);
```

wpModifyStyle Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpModifyStyle Parameter - ulStyleFlags

ulStyleFlags (ULONG) - input
Flag indicating the object's styles to be affected.

Possible values are described in the following list: Values can be combined to affect more than one style in a single call.

OBJSTYLE_NOMOVE
This object cannot move.
OBJSTYLE_LINK
This object cannot have a shadow created.
OBJSTYLE_NOCOPY
This object cannot be copied.
OBJSTYLE_NOTDEFAULTICON
The icon is destroyed when the object goes dormant.
OBJSTYLE_TEMPLATE
This object is a template.
OBJSTYLE_NODELETE
This object cannot be deleted.
OBJSTYLE_NOPRINT
This object cannot be printed.
OBJSTYLE_NODRAG
This object cannot be dragged.
OBJSTYLE_NOTVISIBLE
This object is not visible.
OBJSTYLE_NORENAME
This object cannot be renamed.
OBJSTYLE_NODROP
No other object can be dropped on this object; however, this object can be dragged and dropped on other objects.
OBJSTYLE_NOSETTINGS
This object does not have a Settings view.

wpModifyStyle Parameter - ulStyleMask

ulStyleMask (ULONG) - input
Style flags in *ulStyleFlags* to be set or cleared.

Values specified correspond to the styles specified in *ulStyleFlags* . If the value in *ulStyleMask* is specified (the bit is 1), the corresponding style flag is set. If the value in *ulStyleMask* is not specified (the bit is 0), the corresponding style flag is cleared.

wpModifyStyle Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpModifyStyle - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulStyleFlags (ULONG) - input
Flag indicating the object's styles to be affected.

Possible values are described in the following list: Values can be combined to affect more than one style in a single call.

OBJSTYLE_NOMOVE
This object cannot move.
OBJSTYLE_LINK
This object cannot have a shadow created.
OBJSTYLE_NOCOPY
This object cannot be copied.
OBJSTYLE_NOTDEFAULTICON
The icon is destroyed when the object goes dormant.
OBJSTYLE_TEMPLATE
This object is a template.
OBJSTYLE_NODELETE
This object cannot be deleted.
OBJSTYLE_NOPRINT
This object cannot be printed.
OBJSTYLE_NODRAG
This object cannot be dragged.
OBJSTYLE_NOTVISIBLE
This object is not visible.
OBJSTYLE_NORENAME
This object cannot be renamed.
OBJSTYLE_NODROP
No other object can be dropped on this object; however, this object can be dragged and dropped on other objects.
OBJSTYLE_NOSETTINGS
This object does not have a Settings view.

ulStyleMask (ULONG) - input

Style flags in *ulStyleFlags* to be set or cleared.

Values specified correspond to the styles specified in *ulStyleFlags* . If the value in *ulStyleMask* is specified (the bit is 1), the corresponding style flag is set. If the value in *ulStyleMask* is not specified (the bit is 0), the corresponding style flag is cleared.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpModifyStyle - Remarks

This method sets and clears corresponding style flags as an atomic operation. It can be used in place of the wpQueryStyle and wpSetStyle instance methods.

wpModifyStyle - Usage

This method can be called at any time to change an object's style.

wpModifyStyle - How to Override

This method is generally not overridden.

wpModifyStyle - Related Methods

Related Methods

- wpQueryStyle
- wpSetStyle
- wpclsQueryStyle

wpModifyStyle - Example Code

The following code changes the style of an object to 'not moveable', 'not copyable', 'deleteable'.

```
_wpModifyStyle (self, OBJSTYLE_NOMOVE |
                      OBJSTYLE_NOCOPY |
                      OBJSTYLE_NODELETE,

                      OBJSTYLE_NOMOVE |
                      OBJSTYLE_NOCOPY);
```

wpModifyStyle - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpModuleForClass

wpModuleForClass - Syntax

This method is specific to Version 3, or higher, of the OS/2 operating system.

This instance method returns the module name for a given class.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPClassManager     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
PSZ                 pszClass;   /*  Class for which the module name is returned. */
PSZ                 pszModule;  /*  Module name for pszClass. */

pszModule = _wpModuleForClass(somSelf, pszClass);
```

wpModuleForClass Parameter - somSelf

somSelf (WPClassManager *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class SOMClassMgr.

wpModuleForClass Parameter - pszClass

pszClass (PSZ) - input
Class for which the module name is returned.

wpModuleForClass Return Value - pszModule

pszModule (PSZ) - returns
Module name for *pszClass* .

wpModuleForClass - Parameters

somSelf (WPClassManager *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class SOMClassMgr.

pszClass (PSZ) - input
Class for which the module name is returned.

pszModule (PSZ) - returns
Module name for *pszClass* .

wpModuleForClass - Usage

This method can be called at any time. It is invoked using SOMClassMgrObject.

wpModuleForClass - How to Override

This method should not be overridden.

wpModuleForClass - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Glossary

wpMoveJobObject

wpMoveJobObject - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method moves a print job to a different printer.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPJob          *somSelf;      /*  Pointer to the object on which the method is being invoked. */
WPPrinter      *DestPrinter;  /*  Destination printer. */
ULONG           ulReserved;   /*  Reserved.  Set to 0. */
BOOL            rc;           /*  Success indicator */

rc = _wpMoveJobObject(somSelf, DestPrinter,
       ulReserved);
```

wpMoveJobObject Parameter - somSelf

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

wpMoveJobObject Parameter - DestPrinter

DestPrinter (WPPrinter *) - input
Destination printer.

wpMoveJobObject Parameter - ulReserved

ulReserved (ULONG) - input
Reserved. Set to 0.

wpMoveJobObject Return Value - rc

rc (BOOL) - returns

Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpMoveJobObject - Parameters

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

DestPrinter (WPPrinter *) - input
Destination printer.

ulReserved (ULONG) - input
Reserved. Set to 0.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpMoveJobObject - Related Methods

Related Methods

- wpSetJobProperties

wpMoveJobObject - Topics

Select an item:
Syntax
Parameters
Returns
Related Methods
Glossary

wpMoveObject

wpMoveObject - Syntax

This instance method is called to move the object to a different location.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
WPFolder     *Folder;   /*  Pointer to a Folder object. */
BOOL          rc;       /*  Success indicator. */

rc = _wpMoveObject(somSelf, Folder);
```

wpMoveObject Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpMoveObject Parameter - Folder

Folder (WPFolder *) - input
Pointer to a Folder object.

A pointer to a Folder object into which to move this object. This pointer can be determined by issuing a call to wpclsQueryFolder.

wpMoveObject Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpMoveObject - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

Folder (WPFolder *) - input
Pointer to a Folder object.

A pointer to a Folder object into which to move this object. This pointer can be determined by issuing a call to wpclsQueryFolder.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpMoveObject - Usage

This method can be called at any time in order to move an object to a new location.

wpMoveObject - How to Override

This method is generally not overridden except by storage classes. The parent should be called last unless special actions need to take place when an object is moved.

wpMoveObject - Related Methods

Related Methods

- wpCopyObject
- wpCreateShadowObject

wpMoveObject - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpObjectReady

wpObjectReady - Syntax

This instance method is called as a notification to the object that its creation or awakening is complete.

```
#define INCL_WINWORKPLACE
#include <os2.h>
```

```
WPObject     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
ULONG         ulCode;     /*  Code indicating the type of operation which has just completed. */
WPObject     *refObject;  /*  Pointer to the object from which this object was created. */

wpObjectReady(somSelf, ulCode, refObject);
```

wpObjectReady Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpObjectReady Parameter - ulCode

ulCode (ULONG) - input
Code indicating the type of operation which has just completed.

Possible values are described in the following list:

OR_AWAKE
object awakened from dormancy.
OR_FROMTEMPLATE
Self created from template object *refObject* .
OR_FROMCOPY
Self created as a copy of *refObject* .
OR_NEW
New object created.
OR_SHADOW
Self created as a shadow of *refObject* .
OR_REFERENCE
Set to indicate that the new object (self) was created from the *refObject* .

wpObjectReady Parameter - refObject

refObject (WPObject *) - input
Pointer to the object from which this object was created.

If OR_REFERENCE is set in *ulCode* , this is a pointer to the object from which self was created; otherwise, it is NULL.

wpObjectReady - Return Value

There is no return value for this method.

wpObjectReady - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulCode (ULONG) - input
Code indicating the type of operation which has just completed.

Possible values are described in the following list:

OR_AWAKE
object awakened from dormancy.
OR_FROMTEMPLATE
Self created from template object *refObject*.
OR_FROMCOPY
Self created as a copy of *refObject*.
OR_NEW
New object created.
OR_SHADOW
Self created as a shadow of *refObject*.
OR_REFERENCE
Set to indicate that the new object (self) was created from the *refObject*.

refObject (WPObject *) - input
Pointer to the object from which this object was created.

If OR_REFERENCE is set in *ulCode*, this is a pointer to the object from which self was created; otherwise, it is NULL.

There is no return value for this method.

wpObjectReady - Remarks

The last step of object creation or awakening is the invocation of wpObjectReady. This is a notification to the object indicating completion of the creation or awakening operations.

wpObjectReady - Usage

This method was originally created to indicate when it is safe to call wpSaveDeferred on a newly instantiated object.

wpObjectReady - How to Override

This method is overridden to receive this notification. The parent method should be called in the override to this method.

wpObjectReady - Related Methods

Related Methods

- wpIsObjectInitialized

wpObjectReady - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpOpen

wpOpen - Syntax

This instance method is called to open a view to the object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
HWND          hwndCnr;  /*  Handle of the container window. */
ULONG         ulView;   /*  Specifies which view to open. */
ULONG         param;    /*  Open view parameter. */
HWND          rc;       /*  Success indicator. */

rc = _wpOpen(somSelf, hwndCnr, ulView, param);
```

wpOpen Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpOpen Parameter - hwndCnr

hwndCnr (HWND) - input
Handle of the container window.

Handle of the container window from which the object is opened. This value may be set to NULLHANDLE.

wpOpen Parameter - ulView

ulView (ULONG) - input
Specifies which view to open.

OPEN_CONTENTS
Open contents view.
OPEN_DEFAULT
Open default view (same as double-click).
OPEN_DETAILS
Open details view.
OPEN_HELP
Display HelpPanel.
OPEN_PALETTE
Open palette object.
OPEN_PROMPTDLG
Open prompt dialog.
OPEN_RUNNING
Execute object.
OPEN_SETTINGS
Open Settings notebook.
OPEN_TREE
Open tree view.
OPEN_USER
Class-specific views have a greater value than this.

wpOpen Parameter - param

param (ULONG) - input
Open view parameter.

This value is (reserved = NULL) for views supported by the WPObject class.

wpOpen Return Value - rc

rc (HWND) - returns
Success indicator.

NULLHANDLE	Error occurred.
Other	Handle to either window created or program executed.

wpOpen - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndCnr (HWND) - input
Handle of the container window.

Handle of the container window from which the object is opened. This value may be set to NULLHANDLE.

ulView (ULONG) - input
Specifies which view to open.

OPEN_CONTENTS
Open contents view.
OPEN_DEFAULT
Open default view (same as double-click).
OPEN_DETAILS
Open details view.
OPEN_HELP
Display HelpPanel.
OPEN_PALETTE
Open palette object.
OPEN_PROMPTDLG
Open prompt dialog.
OPEN_RUNNING
Execute object.
OPEN_SETTINGS
Open Settings notebook.
OPEN_TREE
Open tree view.
OPEN_USER
Class-specific views have a greater value than this.

param (ULONG) - input
Open view parameter.

This value is (reserved = NULL) for views supported by the WPObject class.

rc (HWND) - returns
Success indicator.

NULLHANDLE	Error occurred.
Other	Handle to either window created or program executed.

wpOpen - Remarks

In general, wpViewObject should be used instead of this method because wpViewObject takes into consideration the setting in the Object Open Behavior field on the Window page of the Settings notebook for the object. If a view of the object is already open, wpViewObject will, depending on the setting of the Object Open Behavior field, either display the existing window for the object or create a new object.

In contrast, wpOpen *always* opens a new view of the object. Under certain circumstances this might be called for; however, under most circumstances, wpViewObject should be called instead.

wpOpen - Usage

This method can be called at any time in order to open a view of an object.

wpOpen - How to Override

This method should be overridden in order to process class-specific open views. This method can also be overridden in order to modify the

behavior defined by an ancestor class. When wpOpen is overridden to implement a user-defined view, it should call wpAddToObjUseList and wpRegisterView to ensure that the new view is registered by the Workplace Shell as a view of the object.

wpOpen - Related Methods

Related Methods

- wpClose

wpOpen - Example Code

This example performs special processing upon first opening a folder.

```
SOM_Scope HWND   SOMLINK myfold_wpOpen(MYFOLDER *somSelf,
                 HWND hwndCnr,
                 ULONG ulView,
                 ULONG param)

{
    HWND        hwndFromParentOpen = NULLHANDLE;
    MYFOLDERData *somThis = MYFOLDERGetData(somSelf);
    MYFOLDERMethodDebug("MYFOLDER","myfold_wpOpen");

    if (_queryFirstViewHandle(somSelf) == NULLHANDLE)
    {
              /* ONLY do this on FIRST open of the folder. */

       if (_wpPopulate(somSelf, NULLHANDLE, NULL, FALSE))

        {
           WPObject *Obj;
           WPObject *LastFoundObj = NULL;

           /* Check contents of folder and see which are instances of
            * WPFolder, we'll delete any WPFolders as a cleanup step */

           for ( Obj = _wpQueryContent(somSelf,NULL,(ULONG)QC_First);
                 Obj;
                 Obj = _wpQueryContent(somSelf, Obj, (ULONG) QC_Next ))

           {
              /* delete the last object found on the previous iteration
               * of the loop (we couldn't delete it then, because we
               * needed it at the top of the loop for this iteration)
               */
              if (LastFoundObj) {
                 _wpDelete(LastFoundObj,0);
                 LastFoundObj = NULL;
              } /* endif */

              /* we want to make sure that all previously
               * existing folders are deleted */

              if (_somIsA(Obj, _WPFolder)) {
                 LastFoundObj = Obj;
              }  /* endif */

           }  /* endfor */

           /* if there is still one object left to delete, do it now */


           if (LastFoundObj) {
              _wpDelete(LastFoundObj,0);
              LastFoundObj = NULL;
           } /* endif */
       } /* end if populate */
    } /* end if first open */
```

```
            /* Show the opened folder */

    hwndFromParentOpen = parent_wpOpen(somSelf, hwndCnr, ulView, param);

             /* If this is the first open, set our global flag */

    if (_queryFirstViewHandle(somSelf) == NULLHANDLE)  {
       _setFirstViewHandle(somSelf, hwndFromParentOpen);
    } /* endif */

    return( hwndFromParentOpen );
}
```

wpOpen - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpPaintCell

wpPaintCell - Syntax

This instance method is called to paint a cell.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPalette     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PCELL          pCell;    /*  Pointer to the CELL structure to be painted. */
HPS            hps;      /*  Presentation space handle for the cell. */
PRECTL         prcl;     /*  Pointer to the RECTL structure for the area to be painted. */
BOOL           fHilite;  /*  Flag to indicate selected state. */

wpPaintCell(somSelf, pCell, hps, prcl, fHilite);
```

wpPaintCell Parameter - somSelf

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

wpPaintCell Parameter - pCell

pCell (PCELL) - input
Pointer to the CELL structure to be painted.

wpPaintCell Parameter - hps

hps (HPS) - input
Presentation space handle for the cell.

wpPaintCell Parameter - prcl

prcl (PRECTL) - input
Pointer to the RECTL structure for the area to be painted.

wpPaintCell Parameter - fHilite

fHilite (BOOL) - input
Flag to indicate selected state.

TRUE	Cell is currently selected.
FALSE	Cell is not currently selected.

wpPaintCell - Return Value

There is no return value for this method.

wpPaintCell - Parameters

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

pCell (PCELL) - input
Pointer to the CELL structure to be painted.

hps (HPS) - input
Presentation space handle for the cell.

prcl (PRECTL) - input
Pointer to the RECTL structure for the area to be painted.

fHilite (BOOL) - input
Flag to indicate selected state.

TRUE	Cell is currently selected.
FALSE	Cell is not currently selected.

There is no return value for this method.

wpPaintCell - Remarks

The default processing for this method by the WPPalette class is to paint a SYSCLR_WINDOW background. If the cell is highlighted, a SYSCLR_HIGHLITEBACKGROUND background is painted.

wpPaintCell - Usage

This method can be called at any time in order to paint a cell.

wpPaintCell - How to Override

This method should be overriden by all subclasses that want to display visual information in the cell window. It is recommended that the parent method be called first.

wpPaintCell - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Glossary

wpPaintPalette

wpPaintPalette - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to paint the given palette window.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPalette     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
HPS            hps;      /*  Handle to the paint presentation space. */
PRECTL         prcl;     /*  Pointer to the coordinates of the palette window to be painted. */

wpPaintPalette(somSelf, hps, prcl);
```

wpPaintPalette Parameter - somSelf

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

wpPaintPalette Parameter - hps

hps (HPS) - input
Handle to the paint presentation space.

wpPaintPalette Parameter - prcl

prcl (PRECTL) - input
Pointer to the coordinates of the palette window to be painted.

wpPaintPalette - Return Value

There is no return value for this method.

wpPaintPalette - Parameters

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

hps (HPS) - input
Handle to the paint presentation space.

prcl (PRECTL) - input
Pointer to the coordinates of the palette window to be painted.

There is no return value for this method.

wpPaintPalette - How to Override

This method is generally not overridden.

wpPaintPalette - Related Methods

Related Methods

- wpPaintCell

wpPaintPalette - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpPopulate

wpPopulate - Syntax

This instance method is called to allow the folder to populate itself.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;
ULONG         ulReserved;     /*  Reserved value; must be 0. */
PSZ           pszPath;        /*  Reserved value; must be NULL. */
BOOL          fFoldersOnly;   /*  Populate folder. */
BOOL          rc;             /*  Success indicator. */

rc = _wpPopulate(somSelf, ulReserved, pszPath,
       fFoldersOnly);
```

wpPopulate Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpPopulate Parameter - ulReserved

ulReserved (ULONG) - input
Reserved value; must be 0.

wpPopulate Parameter - pszPath

pszPath (PSZ) - input
Reserved value; must be NULL.

wpPopulate Parameter - fFoldersOnly

fFoldersOnly (BOOL) - input
Populate folder.

The *ored* flag indicating the type of contents with which to populate the folder:

TRUE	Populate with folder objects only.
FALSE	Populate with entire contents of folder.

wpPopulate Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpPopulate - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulReserved (ULONG) - input
Reserved value; must be 0.

pszPath (PSZ) - input
Reserved value; must be NULL.

fFoldersOnly (BOOL) - input
Populate folder.

The *ored* flag indicating the type of contents with which to populate the folder:

TRUE	Populate with folder objects only.
FALSE	Populate with entire contents of folder.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpPopulate - Remarks

The wpPopulate method sets the folder flags depending upon the value of fFoldersOnly.

The folder flags indicate the current population state of the folder:

FOI_POPULATEDWITHALL	Folder is completely populated.
FOI_POPULATEDWITHFOLDERS	Folder is populated only with subfolders.

If the folder is re-populated when it has already been populated (as determined by inspecting the folder flags), no action is taken on this message apart from sending the notification message back. This method locks each object it instantiates as a result of the call.

wpPopulate - Usage

This method is generally called only by the system when the folder is opened.

wpPopulate - How to Override

This method can be overridden to alter the contents of a folder. To filter contents added by ancestor classes, it is important to call the parent method first.

wpPopulate - Example Code

This example performs special processing upon first opening a folder.

```
SOM_Scope HWND SOMLINK myfold_wpOpen(MYFOLDER *somSelf,
                                     HWND hwndCnr,
                                     ULONG ulView,
                                     ULONG param)

{
  HWND hwndFromParentOpen = NULLHANDLE;
  MYFOLDERData *somThis = MYFOLDERGetData(somSelf);
  MYFOLDERMethodDebug("MYFOLDER","myfold_wpOpen");

  if (_queryFirstViewHandle(somSelf) == NULLHANDLE)
  {
    /* ONLY do this on FIRST open of the folder */
    if (_wpPopulate(somSelf, NULLHANDLE, NULL, FALSE))
    {
      WPObject *Obj;
      WPObject *LastFoundObj = NULL;
       /* Check contents of folder and see which are instances of  */
       /* WPFolder, we'll delete any WPFolders as a cleanup step   */
       for (Obj = _wpQueryContent(somSelf,NULL,(ULONG)QC_First);
       Obj;
       Obj = _wpQueryContent(somSelf, Obj, (ULONG) QC_Next))

       {
         /* Delete the last object found on the previous iteration */
         /* of the loop.  It couldn't be deleted then, because it  */
         /* was needed at the top of the loop for this iteration.  */
         if (LastFoundObj)
         {
           _wpDelete(LastFoundObj,0);
           LastFoundObj = NULL;
         } /* endif */

         /* Make sure that all previously existing folders */
         /* are deleted                                    */
         if (_somIsA(Obj, _WPFolder))
         {
           LastFoundObj = Obj;
         }  /* endif */
       }  /* endfor */

       /* If there is still one object left to delete, do it now */
       if (LastFoundObj)
       {
         _wpDelete(LastFoundObj,0);
         LastFoundObj = NULL;
       } /* endif */
     } /* end if populate */
   } /* end if first open */

   /* Show the opened folder */
   hwndFromParentOpen = parent_wpOpen(somSelf, hwndCnr, ulView, param);

   /* If this is the first open folder, set the global flag */
   if (_queryFirstViewHandle(somSelf) == NULLHANDLE)
   {
     _setFirstViewHandle(somSelf, hwndFromParentOpen);
   } /* endif */
   return(hwndFromParentOpen);
}
```

wpPopulate - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Glossary

wpPrintJobNext

wpPrintJobNext - Syntax

This instance method is called to print a job next.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPJob     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL       rc;       /*  Success indicator. */

rc = _wpPrintJobNext(somSelf);
```

wpPrintJobNext Parameter - somSelf

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

wpPrintJobNext Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpPrintJobNext - Parameters

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpPrintJobNext - How to Override

This method is generally not overridden.

wpPrintJobNext - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Glossary

wpPrintMetaFile

wpPrintMetaFile - Syntax

This instance method is called to print an object of type MetaFile.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDataFile     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
PPRINTDEST      pPrintDest;  /*  Pointer to the print data. */
BOOL            rc;          /*  Success indicator. */

rc = _wpPrintMetaFile(somSelf, pPrintDest);
```

wpPrintMetaFile Parameter - somSelf

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

wpPrintMetaFile Parameter - pPrintDest

pPrintDest (PPRINTDEST) - input
Pointer to the print data.

It contains all the parameters required to call DevPostDeviceModes and DevOpenDC. For more information about DevPostDeviceModes and DevOpenDC, see the *Presentation Manager Programming Reference* .

wpPrintMetaFile Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpPrintMetaFile - Parameters

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

pPrintDest (PPRINTDEST) - input
Pointer to the print data.

It contains all the parameters required to call DevPostDeviceModes and DevOpenDC. For more information about DevPostDeviceModes and DevOpenDC, see the *Presentation Manager Programming Reference* .

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpPrintMetaFile - How to Override

This method can be overridden by any object class that wants to replace the system-supplied MetaFile print method. It is normally called asynchronously as a result of a call to wpPrintObject.

wpPrintMetaFile - Related Methods

Related Methods

- wpPrintObject

wpPrintMetaFile - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpPrintObject

wpPrintObject - Syntax

This instance method is called to print a view of the object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
PPRINTDEST    pPrintDest;  /*  Pointer to print data. */
ULONG         ulReserved;  /*  Reserved value; must be 0. */
BOOL          rc;          /*  Success indicator. */

rc = _wpPrintObject(somSelf, pPrintDest, ulReserved);
```

wpPrintObject Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpPrintObject Parameter - pPrintDest

pPrintDest (PPRINTDEST) - input
Pointer to print data.

It contains all the parameters required to call DevPostDeviceModes and DevOpenDC. For more information about DevPostDeviceModes and DevOpenDC, see the *Presentation Manager Programming Reference* .

wpPrintObject Parameter - ulReserved

ulReserved (ULONG) - input
Reserved value; must be 0.

wpPrintObject Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpPrintObject - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pPrintDest (PPRINTDEST) - input
Pointer to print data.

It contains all the parameters required to call DevPostDeviceModes and DevOpenDC. For more information about DevPostDeviceModes and DevOpenDC, see the *Presentation Manager Programming Reference* .

ulReserved (ULONG) - input
Reserved value; must be 0.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpPrintObject - Usage

This method can be called at any time in order to print a view of an object.

wpPrintObject - How to Override

This method should be overridden in order to modify the print behavior supported by an ancestor class.

wpPrintObject - Related Methods

Related Methods

- wpPrintMetaFile
- wpPrintPifFile
- wpPrintPlainTextFile
- wpPrintPrinterSpecificFile
- wpPrintUnknownFile

wpPrintObject - Example Code

In this example, where an object does not support being printed, a beep is generated.

```
SOM_Scope BOOL   SOMLINK myfold_wpPrintObject(MYFOLDER *somSelf,
                 PPRINTDEST pPrintDest,
                 ULONG ulReserved)

{
    MYFOLDERData *somThis = MYFOLDERGetData(somSelf);
    MYFOLDERMethodDebug("MYFOLDER","myfold_wpPrintObject");

    DosBeep(100,500);  /* Beep for half a second (500 milliseconds) */

    return( TRUE );
}
```

wpPrintObject - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpPrintPifFile

wpPrintPifFile - Syntax

This instance method is called to print an object of type "Pif".

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDataFile     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
PPRINTDEST      pPrintDest;  /*  Pointer to the print data. */
BOOL            rc;          /*  Success indicator. */

rc = _wpPrintPifFile(somSelf, pPrintDest);
```

wpPrintPifFile Parameter - somSelf

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

wpPrintPifFile Parameter - pPrintDest

pPrintDest (PPRINTDEST) - input
Pointer to the print data.

It contains all the parameters required to call DevPostDeviceModes and DevOpenDC. For more information about DevPostDeviceModes and DevOpenDC, see the *Presentation Manager Programming Reference* .

wpPrintPifFile Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpPrintPifFile - Parameters

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

pPrintDest (PPRINTDEST) - input
Pointer to the print data.

It contains all the parameters required to call DevPostDeviceModes and DevOpenDC. For more information about DevPostDeviceModes and DevOpenDC, see the *Presentation Manager Programming Reference* .

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpPrintPifFile - How to Override

This method can be overridden by any object class that wants to replace the system-supplied Pif file print method. It is normally called asynchronously as a result of a call to wpPrintObject.

wpPrintPifFile - Related Methods

Related Methods

- wpPrintObject

wpPrintPifFile - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpPrintPlainTextFile

wpPrintPlainTextFile - Syntax

This instance method prints a file containing only ASCII text.

```
#define INCL_WINWORKPLACE
#include <os2.h>
```

```
WPFileSystem     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
PPRINTDEST        pPrintDest;  /*  Pointer to the print data. */
BOOL              rc;          /*  Success indicator. */

rc = _wpPrintPlainTextFile(somSelf, pPrintDest);
```

wpPrintPlainTextFile Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpPrintPlainTextFile Parameter - pPrintDest

pPrintDest (PPRINTDEST) - input
Pointer to the print data.

This parameter contains all the parameters required to call DevPostDeviceModes and DevOpenDC. For more information about DevPostDeviceModes and DevOpenDC, see the *Presentation Manager Programming Reference* .

wpPrintPlainTextFile Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpPrintPlainTextFile - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

pPrintDest (PPRINTDEST) - input
Pointer to the print data.

This parameter contains all the parameters required to call DevPostDeviceModes and DevOpenDC. For more information about DevPostDeviceModes and DevOpenDC, see the *Presentation Manager Programming Reference* .

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpPrintPlainTextFile - How to Override

This method can be overridden to replace or remove the system-supplied Plain Text File print algorithm. An override of this method does not call the parent. It is normally called asynchronously as a result of a call to wpPrintObject.

wpPrintPlainTextFile - Related Methods

Related Methods

- wpPrintObject

wpPrintPlainTextFile - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpPrintPrinterSpecificFile

wpPrintPrinterSpecificFile - Syntax

This instance method is called to print an object of type "Printer-specific Data".

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDataFile     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
PPRINTDEST      pPrintDest;  /*  Pointer to the print data. */
BOOL            rc;          /*  Success indicator. */

rc = _wpPrintPrinterSpecificFile(somSelf,
       pPrintDest);
```

wpPrintPrinterSpecificFile Parameter - somSelf

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

wpPrintPrinterSpecificFile Parameter - pPrintDest

pPrintDest (PPRINTDEST) - input
Pointer to the print data.

It contains all the parameters required to call DevPostDeviceModes and DevOpenDC. For more information about DevPostDeviceModes and DevOpenDC, see the *Presentation Manager Programming Reference* .

wpPrintPrinterSpecificFile Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpPrintPrinterSpecificFile - Parameters

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

pPrintDest (PPRINTDEST) - input
Pointer to the print data.

It contains all the parameters required to call DevPostDeviceModes and DevOpenDC. For more information about DevPostDeviceModes and DevOpenDC, see the *Presentation Manager Programming Reference* .

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpPrintPrinterSpecificFile - How to Override

This method can be overridden by any object class that wants to replace the system-supplied Printer-specific File print method. It is normally called asynchronously as a result of a call to wpPrintObject.

wpPrintPrinterSpecificFile - Related Methods

Related Methods

- wpPrintObject

wpPrintPrinterSpecificFile - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpPrintUnknownFile

wpPrintUnknownFile - Syntax

This instance method is called to print an object of "unknown" type.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDataFile     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
PPRINTDEST      pPrintDest;  /*  Pointer to the print data. */
BOOL            rc;          /*  Success indicator. */

rc = _wpPrintUnknownFile(somSelf, pPrintDest);
```

wpPrintUnknownFile Parameter - somSelf

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

wpPrintUnknownFile Parameter - pPrintDest

pPrintDest (PPRINTDEST) - input
Pointer to the print data.

It contains all the parameters required to call DevPostDeviceModes and DevOpenDC. For more information about DevPostDeviceModes and DevOpenDC, see the *Presentation Manager Programming Reference* .

wpPrintUnknownFile Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpPrintUnknownFile - Parameters

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

pPrintDest (PPRINTDEST) - input
Pointer to the print data.

It contains all the parameters required to call DevPostDeviceModes and DevOpenDC. For more information about DevPostDeviceModes and DevOpenDC, see the *Presentation Manager Programming Reference* .

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpPrintUnknownFile - How to Override

This method can be overridden by any object class that wants to replace the system-supplied Unknown File print method. It is normally called asynchronously as a result of a call to wpPrintObject.

wpPrintUnknownFile - Related Methods

Related Methods

- wpPrintObject

wpPrintUnknownFile - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryActionButtons

wpQueryActionButtons - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns a list of action buttons displayed on the Toolbar.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;        /*  Pointer to the object on which the method is being invoked. */
PULONG           pulNumActions;  /*  Pointer to the number of actions returned. */
PACTIONS         pActions;       /*  Pointer to an array of actions displayed on the Toolbar. */

pActions = _wpQueryActionButtons(somSelf,
             pulNumActions);
```

wpQueryActionButtons Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpQueryActionButtons Parameter - pulNumActions

pulNumActions (PULONG) - output
Pointer to the number of actions returned.

wpQueryActionButtons Return Value - pActions

pActions (PACTIONS) - returns
Pointer to an array of actions displayed on the Toolbar.

wpQueryActionButtons - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

pulNumActions (PULONG) - output
Pointer to the number of actions returned.

pActions (PACTIONS) - returns
Pointer to an array of actions displayed on the Toolbar.

wpQueryActionButtons - How to Override

This method can be overridden to add or change actions.

If you want to modify the set of actions returned, call the parent wpQueryActionButtons first.

If you want to return a completely different set of actions, you do not need to call the parent first. If you are adding actions which are not provided or the *ulMenuId* is not a menuID of the WPDesktop class, then you need to also override WPDesktop, using wpMenuItemSelected, to respond to your defined actions.

wpQueryActionButtons - Related Methods

Related Methods

- wpRefreshDrawer

wpQueryActionButtons - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryActionButtonStyle

wpQueryActionButtonStyle - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method determines how the action buttons are displayed on the Toolbar.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG            ulStyle;  /*  Flag indicating the style of the action buttons. */

ulStyle = _wpQueryActionButtonStyle(somSelf);
```

wpQueryActionButtonStyle Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpQueryActionButtonStyle Return Value - ulStyle

ulStyle (ULONG) - returns
Flag indicating the style of the action buttons.

ACTION_BUTTONS_MINI

ACTION_BUTTONS_NORMAL

ACTION_BUTTONS_OFF

ACTION_BUTTONS_TEXT

wpQueryActionButtonStyle - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

ulStyle (ULONG) - returns
Flag indicating the style of the action buttons.

ACTION_BUTTONS_MINI

ACTION_BUTTONS_NORMAL

ACTION_BUTTONS_OFF

ACTION_BUTTONS_TEXT

wpQueryActionButtonStyle - How to Override

This method can be overridden to know when the action display button display style is being queried.

wpQueryActionButtonStyle - Related Methods

Related Methods

- wpRefreshDrawer
- wpSetActionButtonStyle

wpQueryActionButtonStyle - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryAssociatedFileIcon

wpQueryAssociatedFileIcon - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the icon of the program associated with the current data file.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDataFile     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
HPOINTER        hPointer;  /*  Handle of the icon of the program with which the data file is associated. */
```

```
hPointer = _wpQueryAssociatedFileIcon(somSelf);
```

wpQueryAssociatedFileIcon Parameter - somSelf

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

wpQueryAssociatedFileIcon Return Value - hPointer

hPointer (HPOINTER) - returns
Handle of the icon of the program with which the data file is associated.

A return value of NULLHANDLE indicates that there is no associated program icon.

wpQueryAssociatedFileIcon - Parameters

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

hPointer (HPOINTER) - returns
Handle of the icon of the program with which the data file is associated.

A return value of NULLHANDLE indicates that there is no associated program icon.

wpQueryAssociatedFileIcon - Related Methods

Related Methods

- wpQueryAssociatedProgram
- wpSetAssociatedFileIcon

wpQueryAssociatedFileIcon - Topics

Select an item:
Syntax
Parameters
Returns

Related Methods
Glossary

wpQueryAssociatedProgram

wpQueryAssociatedProgram - Syntax

This method is specific to version 2.1, or higher, of the OS/2 operating system.

This instance method returns the WPProgram or WPProgramFile object that is associated with this data file for the specified view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDataFile     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
ULONG           ulView;          /*  View of the data file whose association is being queried. */
PULONG          pulHowMatched;   /*  Flag indicating the association type. */
PSZ             pszMatchString;  /*  Type or name filter. */
ULONG           cbMatchString;   /*  Size of pszMatchString. */
PSZ             pszDefaultType;  /*  Default data type. */
WPObject       *Class;           /*  The WPProgram or WPProgramFile object class associated with this data file

Class = _wpQueryAssociatedProgram(somSelf,
          ulView, pulHowMatched, pszMatchString,
          cbMatchString, pszDefaultType);
```

wpQueryAssociatedProgram Parameter - somSelf

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

wpQueryAssociatedProgram Parameter - ulView

ulView (ULONG) - input
View of the data file whose association is being queried.

This would normally be the result of a _wpQueryDefaultView(somSelf) call.

wpQueryAssociatedProgram Parameter - pulHowMatched

pulHowMatched (PULONG) - input
Flag indicating the association type.

Possible values are described in the following list:

0
Associated by a program added to the menu page.

1
Associated by a match from a name filter.

2
Associated by a match of types.

wpQueryAssociatedProgram Parameter - pszMatchString

pszMatchString (PSZ) - input
Type or name filter.

If *pulHowMatched* is 0, the first byte is set to NULL.

wpQueryAssociatedProgram Parameter - cbMatchString

cbMatchString (ULONG) - input
Size of *pszMatchString* .

wpQueryAssociatedProgram Parameter - pszDefaultType

pszDefaultType (PSZ) - input
Default data type.

Possible values are described in the following list:

string	If the data file has no real type, use this as the default type in the search for matches.
NULL	If the data file has no real type, do not include the system default type in the search for matches.
-1	If the data file has no real type, use the system default type in the search for matches.

wpQueryAssociatedProgram Return Value - Class

Class (WPObject *) - returns
The WPProgram or WPProgramFile object class associated with this data file for the specified view.

A return value of NULL indicates that there is no WPProgram or WPProgramFile object associated for that view.

wpQueryAssociatedProgram - Parameters

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

ulView (ULONG) - input
View of the data file whose association is being queried.

This would normally be the result of a _wpQueryDefaultView(somSelf) call.

pulHowMatched (PULONG) - input
Flag indicating the association type.

Possible values are described in the following list:

0
Associated by a program added to the menu page.

1
Associated by a match from a name filter.

2
Associated by a match of types.

pszMatchString (PSZ) - input
Type or name filter.

If *pulHowMatched* is 0, the first byte is set to NULL.

cbMatchString (ULONG) - input
Size of *pszMatchString* .

pszDefaultType (PSZ) - input
Default data type.

Possible values are described in the following list:

string	If the data file has no real type, use this as the default type in the search for matches.
NULL	If the data file has no real type, do not include the system default type in the search for matches.
-1	If the data file has no real type, use the system default type in the search for matches.

Class (WPObject *) - returns
The WPProgram or WPProgramFile object class associated with this data file for the specified view.

A return value of NULL indicates that there is no WPProgram or WPProgramFile object associated for that view.

wpQueryAssociatedProgram - How to Override

This method must be overriden if your class introduces a new form of association for a range of views not known to the shell.

wpQueryAssociatedProgram - Related Methods

Related Methods

- wpQueryAssociatedFileIcon
- wpSetAssociatedFileIcon

wpQueryAssociatedProgram - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryAssociationFilter

wpQueryAssociationFilter - Syntax

This instance method is called to determine which file title filters are used to associate data file objects to this program object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgram     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PSZ            rc;       /*  Success indicator. */

rc = wpQueryAssociationFilter(somSelf);
```

wpQueryAssociationFilter Parameter - somSelf

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

wpQueryAssociationFilter Return Value - rc

rc (PSZ) - returns
Success indicator.

NULL
Error occurred.

Other
A pointer to a string containing the file title filter. This string can contain several file title filters separated by commas. For example:

```
"*.TXT, *.DOC"
```

wpQueryAssociationFilter - Parameters

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

rc (PSZ) - returns
Success indicator.

NULL
Error occurred.

Other
A pointer to a string containing the file title filter. This string can contain several file title filters separated by commas. For example:

```
"*.TXT, *.DOC"
```

wpQueryAssociationFilter - Remarks

The association filter is used to designate this program as an available open view for data file objects that have a title that matches one of the association filters which are set. If a data file object matches a filter in a program object or program file object, the title of the program object or program file object will appear in the data file object's Open cascade of its pop-up menu. The wpQueryAssociationFilter method will return the filter string set by the last call to the wpSetAssociationFilter method. The wpQueryAssociationType method can be called to determine which file types are used to associate data file objects to this program object.

wpQueryAssociationFilter - Usage

This method can be called at any time in order to determine which file title filters are used to associate data file objects to this program object.

wpQueryAssociationFilter - How to Override

This method is generally not overridden.

wpQueryAssociationFilter - Related Methods

Related Methods

- wpQueryAssociationType
- wpSetAssociationFilter
- wpSetAssociationType

wpQueryAssociationFilter - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpQueryAssociationFilter

wpQueryAssociationFilter - Syntax

This instance method is called to determine which file title filters are used to associate data file objects to this program object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgramFile     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PSZ                rc;       /*  Success indicator. */

rc = _wpQueryAssociationFilter(somSelf);
```

wpQueryAssociationFilter Parameter - somSelf

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

wpQueryAssociationFilter Return Value - rc

rc (PSZ) - returns
Success indicator.

NULL	Error occurred.
Other	A pointer to a string containing the file title filters. This string can contain several file title filters separated by commas. For example:

```
"file*.TXT, *.DOC"
```

wpQueryAssociationFilter - Parameters

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

rc (PSZ) - returns
Success indicator.

NULL	Error occurred.
Other	A pointer to a string containing the file title filters. This string can contain several file title filters separated by commas. For example:

```
"file*.TXT, *.DOC"
```

wpQueryAssociationFilter - Remarks

The association filter is used to designate this program as an available open view for data file objects which have a title that matches one of the association filters which are set. If a data file object matches a filter in a program object or program file object, the title of the program object or program file object will appear in the data file object's Open cascade of its pop-up menu. The wpQueryAssociationFilter method will return the filter string set by the last call to the wpSetAssociationFilter method. The wpQueryAssociationType method can be called to determine which file types are used to associate data file objects to this program object.

wpQueryAssociationFilter - Usage

This method can be called at any time in order to determine which file title filters are used to associate data file objects to this program object.

wpQueryAssociationFilter - How to Override

This method is generally not overridden.

wpQueryAssociationFilter - Related Methods

Related Methods

- wpQueryAssociationType
- wpSetAssociationFilter
- wpSetAssociationType

wpQueryAssociationFilter - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpQueryAssociationType

wpQueryAssociationType - Syntax

This instance method is called to determine which file types are used to associate data file objects to this program object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgram     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PSZ            rc;       /*  Success indicator. */

rc = _wpQueryAssociationType(somSelf);
```

wpQueryAssociationType Parameter - somSelf

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

wpQueryAssociationType Return Value - rc

rc (PSZ) - returns
Success indicator.

NULL	Error occurred.
Other	filedesc. A pointer to a string containing the file type. This string can contain several file title filters separated by commas. For example:

```
"Plain Text,C Code"
```

wpQueryAssociationType - Parameters

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

rc (PSZ) - returns
Success indicator.

NULL	Error occurred.
Other	filedesc. A pointer to a string containing the file type. This string can contain several file title filters separated by commas. For example:

```
"Plain Text,C Code"
```

wpQueryAssociationType - Remarks

The association type is used to designate this program as an available open view for data file objects which have a type that matches one of the association types which are set. If a data file object matches a type in a program object or program file object, the title of the program object or program file object will appear in the data file object's Open cascade of its pop-up menu.

The wpQueryAssociationType method will return the type string set by the last call to the wpSetAssociationType method.

The wpQueryAssociationFilter method can be called to determine which file title filters are used to associate data file objects to this program object.

wpQueryAssociationType - Usage

This method can be called at any time in order to determine which file types are used to associate data file objects to this program object.

wpQueryAssociationType - How to Override

This method is generally not overridden.

wpQueryAssociationType - Related Methods

Related Methods

- wpQueryType
- wpSetAssociationType
- wpSetType

wpQueryAssociationType - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpQueryAssociationType

wpQueryAssociationType - Syntax

This instance method is called to determine which file types are used to associate data file objects to this program object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgramFile     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PSZ                rc;       /*  Success indicator. */

rc = _wpQueryAssociationType(somSelf);
```

wpQueryAssociationType Parameter - somSelf

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

wpQueryAssociationType Return Value - rc

rc (PSZ) - returns
Success indicator.

NULL	Error occurred.
Other	A pointer to a string containing the file types. This string can contain several file title filters separated by commas. For example:

```
"Plain Text,C Code"
```

wpQueryAssociationType - Parameters

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

rc (PSZ) - returns
Success indicator.

NULL	Error occurred.
Other	A pointer to a string containing the file types. This string can contain several file title filters separated by commas. For example:

```
"Plain Text,C Code"
```

wpQueryAssociationType - Remarks

The association type is used to designate this program as an available open view for data file objects which have a type that matches one of the association types which are set. If a data file object matches a type in a program object or program file object, the title of the program object or program file object will appear in the data file object's Open cascade of its pop-up menu. The wpQueryAssociationType method will return the type string set by the last call to the wpSetAssociationType method. The wpQueryAssociationFilter method can be called to determine which file title filters are used to associate data file objects to this program object.

wpQueryAssociationType - Usage

This method can be called at any time in order to determine which file types are used to associate data file objects to this program object.

wpQueryAssociationType - How to Override

This method is generally not overridden.

wpQueryAssociationType - Related Methods

Related Methods

- wpQueryType
- wpSetAssociationType
- wpSetType

wpQueryAssociationType - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpQueryAttr

wpQueryAttr - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method queries the file attributes of a filesystem-based object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG             rc;       /*  File attributes of a filesystem-based object. */

rc = _wpQueryAttr(somSelf);
```

wpQueryAttr Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpQueryAttr Return Value - rc

rc (ULONG) - returns
File attributes of a filesystem-based object.

wpQueryAttr - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

rc (ULONG) - returns
File attributes of a filesystem-based object.

wpQueryAttr - How to Override

This method is generally not overridden.

wpQueryAttr - Related Methods

Related Methods

- wpSetAttr

wpQueryAttr - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryAutoLockup

wpQueryAutoLockup - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns the current value of the automatic lockup option.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL           rc;       /*  Current value of the automatic lockup option. */

rc = _wpQueryAutoLockup(somSelf);
```

wpQueryAutoLockup Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpQueryAutoLockup Return Value - rc

rc (BOOL) - returns
Current value of the automatic lockup option.

TRUE	Automatic lockup selected.
FALSE	Manual lockup selected.

wpQueryAutoLockup - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

rc (BOOL) - returns
Current value of the automatic lockup option.

TRUE	Automatic lockup selected.
FALSE	Manual lockup selected.

wpQueryAutoLockup - Usage

This method can be called at any time in order to determine the automatic lockup option for the current desktop.

wpQueryAutoLockup - How to Override

This method should not be overridden.

wpQueryAutoLockup - Related Methods

Related Methods

- wpQueryLockupAutoDim
- wpQueryLockupBackground
- wpQueryLockupFullScreen
- wpQueryLockupOnStartup
- wpQueryLockupTimeout
- wpSetAutoLockup
- wpSetLockupAutoDim
- wpSetLockupBackground
- wpSetLockupFullScreen
- wpSetLockupOnStartup
- wpSetLockupTimeout

wpQueryAutoLockup - Example Code

In this example, we find the current Desktop object and retrieve the current automatic lockup option setting.

```
WPDesktop    *Desktop;
BOOL         bAutoLockup;

Desktop = _wpclsQueryObjectFromPath("<WP_DESKTOP>");
bAutoLockup = _wpQueryAutoLockup(Desktop);
```

wpQueryAutoLockup - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryAutoRefresh

wpQueryAutoRefresh - Syntax

This method is specific to version 2.1, or higher, of the OS/2 operating system.

This instance method queries the current automatic refresh value for the power object's status window.

```
#define INCL_WINWORKPLACE
#include <os2.h>
```

```
WPPower     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL         rc;       /*  Automatic refresh of the power status window. */

rc = _wpQueryAutoRefresh(somSelf);
```

wpQueryAutoRefresh Parameter - somSelf

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

wpQueryAutoRefresh Return Value - rc

rc (BOOL) - returns
Automatic refresh of the power status window.

TRUE	Automatic refresh of the power status window enabled.
FALSE	Automatic refresh of the power status window disabled.

wpQueryAutoRefresh - Parameters

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

rc (BOOL) - returns
Automatic refresh of the power status window.

TRUE	Automatic refresh of the power status window enabled.
FALSE	Automatic refresh of the power status window disabled.

wpQueryAutoRefresh - Related Methods

Related Methods

- wpAddPowerPage
- wpAddPowerViewPage
- wpChangePowerState
- wpQueryAutoRefresh
- wpQueryPowerConfirmation
- wpQueryPowerManagement
- wpQueryRefreshRate
- wpSetAutoRefresh
- wpSetPowerConfirmation
- wpSetPowerManagement
- wpSetRefreshRate

wpQueryAutoRefresh - Topics

Select an item:
Syntax
Parameters
Returns
Related Methods
Glossary

wpQueryBitmapData

wpQueryBitmapData - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns the pointer to the bitmap data for the image file.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPBitmap     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG        *pulSize;  /*  Pointer to the number of bytes in the bitmap data. */
PBYTE         rc;       /*  Pointer to the bitmap data. */

rc = _wpQueryBitmapData(somSelf, pulSize);
```

wpQueryBitmapData Parameter - somSelf

somSelf (WPBitmap *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPBitmap.

wpQueryBitmapData Parameter - pulSize

pulSize (ULONG *) - in/out
Pointer to the number of bytes in the bitmap data.

wpQueryBitmapData Return Value - rc

rc (PBYTE) - returns
Pointer to the bitmap data.

NULL	Unsuccessful completion.

wpQueryBitmapData - Parameters

somSelf (WPBitmap *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPBitmap.

pulSize (ULONG *) - in/out
Pointer to the number of bytes in the bitmap data.

rc (PBYTE) - returns
Pointer to the bitmap data.

NULL	Unsuccessful completion.

wpQueryBitmapData - Usage

The Workplace Shell calls the wpQueryBitmapData method whenever it needs to access the raw data for a bitmap represented by a WPImageFile object.

wpQueryBitmapData - How to Override

This method must be overridden in any subclass of

wpQueryBitmapData - Related Methods

Related Methods

- wpQueryBitmapHandle
- wpQueryBitmapInfoHeader
- wpReadImageFile
- wpSetBitmapData
- wpWriteImageFile

wpQueryBitmapData - Example Code

This example shows how the WPBitmap class returns the data for a bitmap object. The wpReadImageFile method reads the bitmap data from the data file and sets the instance variables _pBitmapData to point to the bitmap data and _ulBitmapDataSize to the size of the bitmap data.

```
SOM_Scope PBYTE bmp_wpQueryBitmapData(WPBitmap *somSelf,
                                      ULONG *pulSize)
{
    WPBitmapData *somThis = WPBitmapGetData(somSelf);

    /* Read the data from the bitmap file   */
    _wpReadImageFile(somSelf);

    /* Pass the data back to the caller  */
    if(pulSize)
    {
       *pulSize = _ulBitmapDataSize;
    }

    return _pBitmapData;
}
```

wpQueryBitmapData - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryBitmapHandle

wpQueryBitmapHandle - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns a handle to the bitmap that can be displayed showing the contents of the image file and a handle to the palette to be used when displaying the bitmap.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPBitmap     *somSelf;           /*  Pointer to the object on which the method is being invoked. */
HBITMAP      *phBitmap;          /*  Pointer to the buffer to receive the handle of the bitmap. */
HPAL         *phPalette;         /*  Pointer to the buffer to receive the handle of the palette.  If this parame
ULONG         ulWidth;           /*  Width of the bitmap.  This parameter is required.  If this parameter is 0,
ULONG         ulHeight;          /*  Height of the bitmap.  This parameter is required.  If this parameter is 0,
ULONG         ulFlags;           /*  Processing flags. */
LONG          lBackgroundColor;  /*  Color to be used for transparent pixels. */
BOOL         *pbQuitEarly;       /*  Buffer containing quit early flag.  If this parameter is NULL, the caller w
BOOL32        rc;                /*  Success indicator. */

rc = _wpQueryBitmapHandle(somSelf, phBitmap,
```

```
phPalette, ulWidth, ulHeight, ulFlags,
lBackgroundColor, pbQuitEarly);
```

wpQueryBitmapHandle Parameter - somSelf

somSelf (WPBitmap *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPBitmap.

wpQueryBitmapHandle Parameter - phBitmap

phBitmap (HBITMAP *) - in/out
Pointer to the buffer to receive the handle of the bitmap.

wpQueryBitmapHandle Parameter - phPalette

phPalette (HPAL *) - in/out
Pointer to the buffer to receive the handle of the palette. If this parameter is NULL, no palette handle is returned.

wpQueryBitmapHandle Parameter - ulWidth

ulWidth (ULONG) - input
Width of the bitmap. This parameter is required. If this parameter is 0, the bitmap will not be scaled.

wpQueryBitmapHandle Parameter - ulHeight

ulHeight (ULONG) - input
Height of the bitmap. This parameter is required. If this parameter is 0, the bitmap will not be scaled.

wpQueryBitmapHandle Parameter - ulFlags

ulFlags (ULONG) - input
Processing flags.

wpQueryBitmapHandle Parameter - lBackgroundColor

lBackgroundColor (LONG) - input
Color to be used for transparent pixels.

wpQueryBitmapHandle Parameter - pbQuitEarly

pbQuitEarly (BOOL *) - in/out
Buffer containing quit early flag. If this parameter is NULL, the caller will not be able to asynchronously abort the loading of the bitmap.

wpQueryBitmapHandle Return Value - rc

rc (BOOL32) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryBitmapHandle - Parameters

somSelf (WPBitmap *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPBitmap.

phBitmap (HBITMAP *) - in/out
Pointer to the buffer to receive the handle of the bitmap.

phPalette (HPAL *) - in/out
Pointer to the buffer to receive the handle of the palette. If this parameter is NULL, no palette handle is returned.

ulWidth (ULONG) - input
Width of the bitmap. This parameter is required. If this parameter is 0, the bitmap will not be scaled.

ulHeight (ULONG) - input
Height of the bitmap. This parameter is required. If this parameter is 0, the bitmap will not be scaled.

ulFlags (ULONG) - input
Processing flags.

lBackgroundColor (LONG) - input
Color to be used for transparent pixels.

pbQuitEarly (BOOL *) - in/out

Buffer containing quit early flag. If this parameter is NULL, the caller will not be able to asynchronously abort the loading of the bitmap.

rc (BOOL32) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryBitmapHandle - Usage

The Workplace Shell calls the wpQueryBitmapHandle method any time it needs to access the bitmap represented by a WPImageFile object.

wpQueryBitmapHandle - How to Override

This method must be overridden in any subclass of WPImageFile

wpQueryBitmapHandle - Related Methods

Related Methods

- wpQueryBitmapData
- wpQueryBitmapInfoHeader
- wpReadImageFile
- wpSetBitmapData
- wpWriteImageFile

wpQueryBitmapHandle - Example Code

This example is a simplified version of the code in the WPBitmap class. The example does build a color palette for the bitmap and does no scaling of the bitmap size. (In this code, vhab is a global variable that contains the handle for the HAB for the Workplace Shell and RemoveBitmapFromMem is a function that frees the specified Device Context, Presentation Space, Bitmap, and Palette.)

```
SOM_Scope BOOL32 bmp_wpQueryBitmapHandle(WPBitmap *somSelf,
                                         HBITMAP *phBitmap,
                                         HPAL *phPalette,
                                         ULONG ulWidth,
                                         ULONG ulHeight,
                                         ULONG ulFlags,
                                         LONG lBackgroundColor,
                                         BOOL *pbQuitEarly)
{
   HDC         hdc = NULLHANDLE;
   HPS         hps = NULLHANDLE;
   union
   {
      PBITMAPFILEHEADER  pbfh;
      PBITMAPFILEHEADER2 pbfh2;
   } pBitmapFileHeader;
   union
   {
      PBITMAPINFOHEADER  pbinfoh;
      PBITMAPINFOHEADER2 pbinfoh2;
      PBITMAPINFO2       pbinfo2;
   } pBitmapInfoHeader;
   SIZEL       sizlSource;
```

```
   /* Make sure the caller specified a place to return the bitmap handle */
   if (!phBitmap)
   {
      return FALSE;
   }

   /* Set the returned bitmap and palette handles to NULLHANDLE in case
    * this method fails  */
   *phBitmap = NULLHANDLE;
   if (phPalette)
   {
      *phPalette = NULLHANDLE;
   }

   /* Read the bitmap file   */
   if (!_wpReadImageFile(somSelf))
   {
      return FALSE;
   }

   /* Create a device context and presentation space for the bitmap  */
   hdc = DevOpenDC (vhab, OD_MEMORY, "*", 0L, NULL, NULLHANDLE);
   if (hdc == DEV_ERROR)
   {
      return FALSE;
   }

   /*
    *  Create a new Presentation Space for the bitmap  */
   sizlSource.cx = 0;
   sizlSource.cy = 0;
   hps = GpiCreatePS (vhab,
                      hdc,
                      &sizlSource,
                      PU_PELS | GPIT_NORMAL | GPIA_ASSOC);
   if(hps == GPI_ERROR)
   {
      RemoveBitmapFromMem(&hdc,NULL,NULL,NULL);
      return FALSE;
   }

   /* Make sure there is no palette selected in the PS   */
   GpiSelectPalette(hps,NULLHANDLE);

   /* Create the bitmap   */
   pBitmapFileHeader.pbfh = (PBITMAPFILEHEADER)
                            _wpQueryBitmapData(somSelf,NULL);
   pBitmapInfoHeader.pbinfoh = &(pBitmapFileHeader.pbfh->bmp);
   *phBitmap = GpiCreateBitmap (hps,
                                pBitmapInfoHeader.pbinfoh2,
                                CBM_INIT,
                                (PBYTE)pBitmapFileHeader.pbfh +
                                   pBitmapFileHeader.pbfh->offBits,
                                pBitmapInfoHeader.pbinfo2);
   if(!*phBitmap)
   {
      RemoveBitmapFromMem(&hdc,&hps,NULL,NULL);
      return FALSE;
   }

   /*
    *  Select the bitmap into the presentation space  */
   GpiSetBitmap (hps, *phBitmap);

   /* Clean up the resources we allocated   */
   RemoveBitmapFromMem(&hdc,&hps,NULL,NULL);
   return !!*phBitmap;
}
```

wpQueryBitmapHandle - Topics

Select an item:
Syntax

Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryBitmapInfoHeader

wpQueryBitmapInfoHeader - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns the pointer to the bitmap information header for the image file.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPBitmap     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PBYTE         rc;       /*  The bitmap information header structure. */

rc = _wpQueryBitmapInfoHeader(somSelf);
```

wpQueryBitmapInfoHeader Parameter - somSelf

somSelf (WPBitmap *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPBitmap.

wpQueryBitmapInfoHeader Return Value - rc

rc (PBYTE) - returns
The bitmap information header structure.

NonNULL	Successful completion.
NULL	Error occurred.

wpQueryBitmapInfoHeader - Parameters

somSelf (WPBitmap *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPBitmap.

rc (PBYTE) - returns
The bitmap information header structure.

NonNULL	Successful completion.
NULL	Error occurred.

wpQueryBitmapInfoHeader - Usage

The Workplace Shell calls the wpQueryBitmapInfoHeader function anytime it needs to examine the header for the bitmap represented by a WPImageFile object. For example, it can call the wpQueryBitmapInfoHeader method to find out the size of the bitmap when calculating the characteristics needed to tile the bitmap for the background of a folder.

wpQueryBitmapInfoHeader - How to Override

This method must be overridden in any subclass of WPImageFile.

wpQueryBitmapInfoHeader - Related Methods

Related Methods

- wpQueryBitmapData
- wpQueryBitmapHandle
- wpReadImageFile
- wpSetBitmapData
- wpWriteImageFile

wpQueryBitmapInfoHeader - Example Code

This example shows how the WPBitmap class returns the bitmap information header.

```
SOM_Scope PBYTE bmp_wpQueryBitmapInfoHeader(WPBitmap *somSelf)
{
    PBITMAPFILEHEADER2  pbfh2;
    ULONG               ulBitmapDataSize;

    pbfh2 =
    (PBITMAPFILEHEADER2)_wpQueryBitmapData(somSelf,&ulBitmapDataSize);
    return (ulBitmapDataSize ?
            (PBYTE)&(pbfh2->bmp2) :
            NULL);
}
```

wpQueryBitmapInfoHeader - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryButtonAppearance

wpQueryButtonAppearance - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the Frame Control Button appearance of an object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
ULONG         ulButtonType;  /*  A flag indicating the type of control button. */

ulButtonType = _wpQueryButtonAppearance(somSelf);
```

wpQueryButtonAppearance Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryButtonAppearance Return Value - ulButtonType

ulButtonType (ULONG) - returns
A flag indicating the type of control button.

Possible values are described in the following list:

HIDEBUTTON

MINBUTTON

DEFAULTBUTTON

wpQueryButtonAppearance - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulButtonType (ULONG) - returns
A flag indicating the type of control button.

Possible values are described in the following list:

HIDEBUTTON

MINBUTTON

DEFAULTBUTTON

wpQueryButtonAppearance - Usage

This method is usually called by the system to retrieve the object's Frame Control Button appearance.

wpQueryButtonAppearance - How to Override

This method is generally not overridden.

wpQueryButtonAppearance - Related Methods

Related Methods

- wpSetButtonAppearance
- wpclsQueryButtonAppearance

wpQueryButtonAppearance - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpQueryCloseDrawer

wpQueryCloseDrawer - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method determines whether the drawer closes after an object in the drawer is opened from the drawer.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL             fState;   /*  Flag indicating whether to close the drawer. */

fState = _wpQueryCloseDrawer(somSelf);
```

wpQueryCloseDrawer Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpQueryCloseDrawer Return Value - fState

fState (BOOL) - returns
Flag indicating whether to close the drawer.

TRUE
The drawer closes.

FALSE
The drawer does not close.

wpQueryCloseDrawer - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

fState (BOOL) - returns
Flag indicating whether to close the drawer.

TRUE
The drawer closes.
FALSE
The drawer does not close.

wpQueryCloseDrawer - How to Override

This method can be overridden to force a particular value.

wpQueryCloseDrawer - Related Methods

Related Methods

- wpSetCloseDrawer

wpQueryCloseDrawer - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryComputerName

wpQueryComputerName - Syntax

This instance method is called to query the name of the computer on which the printer object exists.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPrinter     *somSelf;          /*  Pointer to the object on which the method is being invoked. */
PSZ            pszComputerName;  /*  Returned computer name of the object queried. */
ULONG          rc;               /*  Success indicator. */

rc = _wpQueryComputerName(somSelf, pszComputerName);
```

wpQueryComputerName Parameter - somSelf

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

wpQueryComputerName Parameter - pszComputerName

pszComputerName (PSZ) - output
Returned computer name of the object queried.

wpQueryComputerName Return Value - rc

rc (ULONG) - returns
Success indicator.

0	An error occurred.
1	Successful - ComputerName is local (NULL).
2	Successful - ComputerName is on a network. See *pszComputerName* for values.

wpQueryComputerName - Parameters

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

pszComputerName (PSZ) - output
Returned computer name of the object queried.

rc (ULONG) - returns
Success indicator.

0	An error occurred.
1	Successful - ComputerName is local (NULL).
2	Successful - ComputerName is on a network. See *pszComputerName* for values.

wpQueryComputerName - How to Override

This method is generally not overridden.

wpQueryComputerName - Related Methods

Related Methods

- wpQueryPrinterName
- wpSetComputerName

wpQueryComputerName - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryConcurrentView

wpQueryConcurrentView - Syntax

This instance method is called to query the multiple, concurrent view behavior of an object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         ulView;   /*  Flag indicating the concurrent view mode. */

ulView = _wpQueryConcurrentView(somSelf);
```

wpQueryConcurrentView Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryConcurrentView Return Value - ulView

ulView (ULONG) - returns
Flag indicating the concurrent view mode.

Possible values are described in the following list:

CCVIEW_ON
Concurrent view mode is set to on.
CCVIEW_OFF
Concurrent view mode is set to off.
CCVIEW_DEFAULT
Concurrent view mode is set to the system default.

wpQueryConcurrentView - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulView (ULONG) - returns
Flag indicating the concurrent view mode.

Possible values are described in the following list:

CCVIEW_ON
Concurrent view mode is set to on.
CCVIEW_OFF
Concurrent view mode is set to off.
CCVIEW_DEFAULT
Concurrent view mode is set to the system default.

wpQueryConcurrentView - How to Override

This method should not be overridden.

wpQueryConcurrentView - Related Methods

Related Methods

- wpSetConcurrentView

wpQueryConcurrentView - Example Code

This example turns on concurrent views for MYFOLDER by default.

```
SOM_Scope ULONG  SOMLINK myfold_wpQueryConcurrentView(MYFOLDER *somSelf )
```

```
{
/* MYFOLDERData *somThis = MYFOLDERGetData(somSelf); */
   MYFOLDERMethodDebug("MYFOLDER","myfold_wpQueryConcurrentView");

   return CCVIEW_ON;
}
```

wpQueryConcurrentView - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Example Code
Related Methods
Glossary

wpQueryConfirmations

wpQueryConfirmations - Syntax

This instance method is called to determine which confirmations are set on this object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         ulflag;   /*  Confirmation flags. */

ulflag = _wpQueryConfirmations(somSelf);
```

wpQueryConfirmations Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryConfirmations Return Value - ulflag

ulflag (ULONG) - returns
Confirmation flags.

NULL
No confirmations set.

CONFIRM_DELETE
Prompt for confirmation for all objects.

CONFIRM_DELETEFOLDER
Prompt for confirmation for folder objects, only.

CONFIRM_RENAMEFILESWITHEXT
Prompt for confirmation when renaming files with extensions.

For example, when the user renames a file with an EXE extension, the file will no longer be executable. The user should be prompted to confirm the name change.

CONFIRM_KEEPASSOC
Prompt for confirmation when renaming files with extensions that are used to associate a file with an application.

For example, if the user has an association between an editor and the file extension DOC, and the user renames a DOC file to have a TXT extension, the file is no longer associated with the editor. The user should be prompted to confirm the name change.

CONFIRM_ACTION
Prompt for confirmation for copy and move operations and for creating shadows.

CONFIRM_PROGRESS
Display a dialog box that shows the progress being made in a copy, move, or create-shadow operation. The dialog box also has a Pause button that the user can use to control the progress of the operation.

CONFIRM_ALLOWFORCEDELETE
Allow forced deletion; must be set if CONFIRM_FORCEDELETEis set.

CONFIRM_FORCEDELETE
Prompt for confirmation of forced deletion.

wpQueryConfirmations - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulflag (ULONG) - returns
Confirmation flags.

NULL
No confirmations set.

CONFIRM_DELETE
Prompt for confirmation for all objects.

CONFIRM_DELETEFOLDER
Prompt for confirmation for folder objects, only.

CONFIRM_RENAMEFILESWITHEXT
Prompt for confirmation when renaming files with extensions.

For example, when the user renames a file with an EXE extension, the file will no longer be executable. The user should be prompted to confirm the name change.

CONFIRM_KEEPASSOC
Prompt for confirmation when renaming files with extensions that are used to associate a file with an application.

For example, if the user has an association between an editor and the file extension DOC, and the user renames a DOC file to have a TXT extension, the file is no longer associated with the editor. The user should be prompted to confirm the name change.

CONFIRM_ACTION
: Prompt for confirmation for copy and move operations and for creating shadows.

CONFIRM_PROGRESS
: Display a dialog box that shows the progress being made in a copy, move, or create-shadow operation. The dialog box also has a Pause button that the user can use to control the progress of the operation.

CONFIRM_ALLOWFORCEDELETE
: Allow forced deletion; must be set if CONFIRM_FORCEDELETEis set.

CONFIRM_FORCEDELETE
: Prompt for confirmation of forced deletion.

wpQueryConfirmations - Usage

This method can be called at any time in order to determine the confirmations set on an object.

wpQueryConfirmations - How to Override

This method is generally not overridden.

wpQueryConfirmations - Related Methods

Related Methods

- wpConfirmDelete
- wpDelete
- wpFree
- wpQueryConfirmations

wpQueryConfirmations - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpQueryContainerFlagPtr

wpQueryContainerFlagPtr - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns a pointer to the flag indicating whether or not an object is in a container.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;          /*  Pointer to the object on which the method is being invoked. */
PULONG        pFlagIndication;  /*  Pointer to the flag indicating whether or not the object has been inserted i

pFlagIndication = _wpQueryContainerFlagPtr(
                   somSelf);
```

wpQueryContainerFlagPtr Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryContainerFlagPtr Return Value - pFlagIndication

pFlagIndication (PULONG) - returns
Pointer to the flag indicating whether or not the object has been inserted into a container.

wpQueryContainerFlagPtr - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pFlagIndication (PULONG) - returns
Pointer to the flag indicating whether or not the object has been inserted into a container.

wpQueryContainerFlagPtr - Usage

This method can be called at any time to query the pointer to the flag.

wpQueryContainerFlagPtr - How to Override

This method is generally not overridden.

wpQueryContainerFlagPtr - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Glossary

wpQueryContent

wpQueryContent - Syntax

This instance method is called to allow the folder to specify its contents.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
WPObject     *Object;    /*  Pointer to a workplace object. */
ULONG         ulOption;  /*  Flag indicating the object to query (QC_FIRST, QC_NEXT,QC_LAST). */
WPObject     *item;      /*  Pointer to the correct item in the folder's content list. */

item = _wpQueryContent(somSelf, Object, ulOption);
```

wpQueryContent Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryContent Parameter - Object

Object (WPObject *) - input
Pointer to a workplace object.

This field is ignored unless QC_NEXT is specified in ulOption.

wpQueryContent Parameter - ulOption

ulOption (ULONG) - input
Flag indicating the object to query (QC_FIRST, QC_NEXT,QC_LAST).

wpQueryContent Return Value - item

item (WPObject *) - returns
Pointer to the correct item in the folder's content list.

wpQueryContent - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

Object (WPObject *) - input
Pointer to a workplace object.

This field is ignored unless QC_NEXT is specified in ulOption.

ulOption (ULONG) - input
Flag indicating the object to query (QC_FIRST, QC_NEXT,QC_LAST).

item (WPObject *) - returns
Pointer to the correct item in the folder's content list.

wpQueryContent - Remarks

This method allows the user to query the folder's content in various ways using the ulOption flag. QC_FIRST returns the first item in the content list, QC_LAST returns the last item, and QC_NEXT returns the next item after "Object" in the list.

wpQueryContent - Usage

This method is generally called to look for a specific object or to query the contents in a specific folder.

wpQueryContent - How to Override

This method is generally not overridden.

wpQueryContent - Example Code

This example performs special processing upon first opening a folder.

```
SOM_Scope HWND   SOMLINK myfold_wpOpen(MYFOLDER *somSelf,
                 HWND hwndCnr,
                 ULONG ulView,
                 ULONG param)

{
    HWND        hwndFromParentOpen = NULLHANDLE;
    MYFOLDERData *somThis = MYFOLDERGetData(somSelf);
    MYFOLDERMethodDebug("MYFOLDER","myfold_wpOpen");

    if (_queryFirstViewHandle(somSelf) == NULLHANDLE)
    {
              /* ONLY do this on FIRST open of the folder. */

       if (_wpPopulate(somSelf, NULLHANDLE, NULL, FALSE))
        {
           WPObject *Obj;
           WPObject *LastFoundObj = NULL;

           /* Check contents of folder and see which are instances of
            * WPFolder, we'll delete any WPFolders as a cleanup step */

           for ( Obj = _wpQueryContent(somSelf,NULL,(ULONG)QC_First);
                 Obj;
                 Obj = _wpQueryContent(somSelf, Obj, (ULONG) QC_Next ))

           {
              /* delete the last object found on the previous iteration
               * of the loop (we couldn't delete it then, because we
               * needed it at the top of the loop for this iteration)
               */
              if (LastFoundObj) {
                 _wpDelete(LastFoundObj,0);
                 LastFoundObj = NULL;
              } /* endif */

              /* we want to make sure that all previously
               * existing folders are deleted */

              if (_somIsA(Obj, _WPFolder)) {
                 LastFoundObj = Obj;
              }  /* endif */

           }  /* endfor */

           /* if there is still one object left to delete, do it now */


           if (LastFoundObj) {
              _wpDelete(LastFoundObj,0);
              LastFoundObj = NULL;
           } /* endif */
       } /* end if populate */
    } /* end if first open */

              /* Show the opened folder */

    hwndFromParentOpen = parent_wpOpen(somSelf, hwndCnr, ulView, param);

               /* If this is the first open, set our global flag */

    if (_queryFirstViewHandle(somSelf) == NULLHANDLE)  {
       _setFirstViewHandle(somSelf, hwndFromParentOpen);
```

```
    } /* endif */

    return( hwndFromParentOpen );
}
```

wpQueryContent - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Glossary

wpQueryCoreRecord

wpQueryCoreRecord - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns a pointer to a MINIRECORDCORE structure associated with a given object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject          *somSelf;      /*  Pointer to the object on which the method is being invoked. */
PMINIRECORDCORE    pMiniRecord;  /*  Pointer to an object's container control record. */

pMiniRecord = _wpQueryCoreRecord(somSelf);
```

wpQueryCoreRecord Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryCoreRecord Return Value - pMiniRecord

pMiniRecord (PMINIRECORDCORE) - returns
Pointer to an object's container control record.

wpQueryCoreRecord - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pMiniRecord (PMINIRECORDCORE) - returns
Pointer to an object's container control record.

wpQueryCoreRecord - Remarks

When an object is created, the Workplace Shell automatically allocates memory for a MINIRECORDCORE structure and stores a pointer to it in the object's instance data. Every object has an associated MINIRECORDCORE structure so it can be placed into the container. This method is used to obtain a pointer to the MINIRECORDCORE structure when the pointer to the object is known.

wpQueryCoreRecord - How to Override

This method is generally not overridden.

wpQueryCoreRecord - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Glossary

wpQueryCreation

wpQueryCreation - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the creation date and time of a file.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
FDATE            *fdate;    /*  Pointer to the date structure in which the file's creation date is returned. */
FTIME            *ftime;    /*  Pointer to the time structure in which the file's creation time is returned. */
ULONG             rc;       /*  Success indicator. */

rc = _wpQueryCreation(somSelf, fdate, ftime);
```

wpQueryCreation Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpQueryCreation Parameter - fdate

fdate (FDATE *) - output
Pointer to the date structure in which the file's creation date is returned.

wpQueryCreation Parameter - ftime

ftime (FTIME *) - output
Pointer to the time structure in which the file's creation time is returned.

wpQueryCreation Return Value - rc

rc (ULONG) - returns
Success indicator.

0
No error.

other
Error occurred.

wpQueryCreation - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

fdate (FDATE *) - output
Pointer to the date structure in which the file's creation date is returned.

ftime (FTIME *) - output
Pointer to the time structure in which the file's creation time is returned.

rc (ULONG) - returns
Success indicator.

0
No error.
other
Error occurred.

wpQueryCreation - How to Override

This method is generally not overridden.

wpQueryCreation - Related Methods

Related Methods

- wpQueryLastAccess
- wpQueryLastWrite

wpQueryCreation - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryDateInfo

wpQueryDateInfo - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the date information from the given file object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
FILEFINDBUF4     *pstFileFindBuf;  /*  Pointer to a buffer of file information. */
ULONG             rc;              /*  Always 0. */

rc = _wpQueryDateInfo(somSelf, pstFileFindBuf);
```

wpQueryDateInfo Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpQueryDateInfo Parameter - pstFileFindBuf

pstFileFindBuf (FILEFINDBUF4 *) - in/out
Pointer to a buffer of file information.

The date and time of creation, last access, and last write are returned in the buffer.

wpQueryDateInfo Return Value - rc

rc (ULONG) - returns
Always 0.

wpQueryDateInfo - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

pstFileFindBuf (FILEFINDBUF4 *) - in/out
Pointer to a buffer of file information.

The date and time of creation, last access, and last write are returned in the buffer.

rc (ULONG) - returns
Always 0.

wpQueryDateInfo - How to Override

This method is generally not overridden.

wpQueryDateInfo - Related Methods

Related Methods

- wpSetDateInfo

wpQueryDateInfo - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryDefaultDropOp

wpQueryDefaultDropOp - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method queries the default drop operation of objects that are dropped on the Desktop. The default is set on the Desktop page of the Desktop's Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG          rc;       /*  Default drop operation. */

rc = _wpQueryDefaultDropOp(somSelf);
```

wpQueryDefaultDropOp Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpQueryDefaultDropOp Return Value - rc

rc (ULONG) - returns
Default drop operation.

DO_LINK	Create shadow of dropped object
DO_MOVE	Move dropped object

wpQueryDefaultDropOp - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

rc (ULONG) - returns
Default drop operation.

DO_LINK	Create shadow of dropped object
DO_MOVE	Move dropped object

wpQueryDefaultDropOp - Usage

This method can be called at any time in order to determine the default drop operation for the current desktop.

wpQueryDefaultDropOp - How to Override

This method can be overridden to control the Desktop's default drop operation

wpQueryDefaultDropOp - Related Methods

Related Methods

- wpSetDefaultDropOp

wpQueryDefaultDropOp - Example Code

In this example, the current Desktop object is found and then the default drop operation for the desktop is done.

```
  WPDesktop    *Desktop;
  ULONG        ulDropOp;

  Desktop  = _wpclsQueryObjectFromPath("<WP_DESKTOP>");
  ulDropOp = _wpQueryDefaultDropOp(Desktop);
  /* Value of ulDropOp is DO_LINK or DO_MOVE */
```

wpQueryDefaultDropOp - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryDefaultHelp

wpQueryDefaultHelp - Syntax

This instance method is called to allow the object to specify its default help panel.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
PULONG        pHelpPanelId;  /*  Pointer to the help panel ID. */
PSZ           HelpLibrary;   /*  Pointer to the help library. */
BOOL          rc;            /*  Success indicator. */

rc = _wpQueryDefaultHelp(somSelf, pHelpPanelId,
       HelpLibrary);
```

wpQueryDefaultHelp Parameter - somSelf

somSelf (WPObject *) - input

Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryDefaultHelp Parameter - pHelpPanelId

pHelpPanelId (PULONG) - output
Pointer to the help panel ID.

wpQueryDefaultHelp Parameter - HelpLibrary

HelpLibrary (PSZ) - output
Pointer to the help library.

The pointer to a buffer in which to place the name of the help library. This buffer should be the length of CCHMAXPATH bytes.

wpQueryDefaultHelp Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryDefaultHelp - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pHelpPanelId (PULONG) - output
Pointer to the help panel ID.

HelpLibrary (PSZ) - output
Pointer to the help library.

The pointer to a buffer in which to place the name of the help library. This buffer should be the length of CCHMAXPATH bytes.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryDefaultHelp - Remarks

The default help panel for this class can be determined by calling the wpclsQueryDefaultHelp method.

wpQueryDefaultHelp - Usage

This method can be called at any time in order to determine the default panel for this object.

wpQueryDefaultHelp - How to Override

This method is generally not overridden.

wpQueryDefaultHelp - Related Methods

Related Methods

- wpDisplayHelp
- wpMenuItemHelpSelected
- wpQueryDefaultHelp
- wpSetDefaultHelp
- wpclsQueryDefaultHelp

wpQueryDefaultHelp - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpQueryDefaultIconPos

wpQueryDefaultIconPos - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the default icon position of the current object within a folder.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PPOINTL       pPointl;  /*  Pointer to the coordinates of the default icon position. */
BOOL          rc;       /*  Success indicator. */

rc = _wpQueryDefaultIconPos(somSelf, pPointl);
```

wpQueryDefaultIconPos Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryDefaultIconPos Parameter - pPointl

pPointl (PPOINTL) - output
Pointer to the coordinates of the default icon position.

The x and y values represent the position in the object's folder in percentage coordinates.

wpQueryDefaultIconPos Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpQueryDefaultIconPos - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pPointl (PPOINTL) - output

Pointer to the coordinates of the default icon position.

The *x* and *y* values represent the position in the object's folder in percentage coordinates.

rc (BOOL) - returns
Success indicator.

TRUE
: Successful completion.

FALSE
: Error occurred.

wpQueryDefaultIconPos - Remarks

Refer to wpSetup for more information about the initial icon position (ICONPOS).

wpQueryDefaultIconPos - Usage

This method can be called at any time to return the default icon position within a folder.

wpQueryDefaultIconPos - How to Override

This method is generally not overridden.

wpQueryDefaultIconPos - Related Methods

Related Methods

- wpSetDefaultIconPos

wpQueryDefaultIconPos - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpQueryDefaultView

wpQueryDefaultView - Syntax

This instance method is called to allow the object to query its current default open view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         ulView;   /*  Flag indicating the default open view. */

ulView = _wpQueryDefaultView(somSelf);
```

wpQueryDefaultView Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryDefaultView Return Value - ulView

ulView (ULONG) - returns
Flag indicating the default open view.

OPEN_CONTENTS
: Open contents view.

OPEN_DEFAULT
: Open default view (same as double-click).

OPEN_DETAILS
: Open details view.

OPEN_HELP
: Display HelpPanel.

OPEN_RUNNING
: Execute object.

OPEN_SETTINGS
: Open Settings notebook.

OPEN_TREE
: Open tree view.

OPEN_UNKNOWN
: Unknown view.

OPEN_USER
: Class-specific views have a greater value than this.

wpQueryDefaultView - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulView (ULONG) - returns
Flag indicating the default open view.

OPEN_CONTENTS
Open contents view.
OPEN_DEFAULT
Open default view (same as double-click).
OPEN_DETAILS
Open details view.
OPEN_HELP
Display HelpPanel.
OPEN_RUNNING
Execute object.
OPEN_SETTINGS
Open Settings notebook.
OPEN_TREE
Open tree view.
OPEN_UNKNOWN
Unknown view.
OPEN_USER
Class-specific views have a greater value than this.

wpQueryDefaultView - Remarks

This method returns the default open view for this instance. The default open view is displayed when a user double-clicks on the object or selects *Open* without selecting an item in the open cascade.

wpQueryDefaultView - Usage

This method can be called at any time in order to determine its default open view.

wpQueryDefaultView - How to Override

This method is generally not overridden.

wpQueryDefaultView - Related Methods

Related Methods

- wpQueryDefaultView
- wpclsQueryDefaultView
- wpSetDefaultView

wpQueryDefaultView - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpQueryDefStatusView

wpQueryDefStatusView - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the default status view of the current power object when Advanced Power Management (APM) is enabled.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPower     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
ULONG        ulDefStatus;  /*  Flag indicating the default status view. */

ulDefStatus = _wpQueryDefStatusView(somSelf);
```

wpQueryDefStatusView Parameter - somSelf

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

wpQueryDefStatusView Return Value - ulDefStatus

ulDefStatus (ULONG) - returns
Flag indicating the default status view.

Possible values are described in the following list:

OPEN_STATUS
The full status, including the battery life, power source, and battery state, is shown.
OPEN_BATTERY
Only the battery life indicator is shown.

wpQueryDefStatusView - Parameters

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

ulDefStatus (ULONG) - returns
Flag indicating the default status view.

Possible values are described in the following list:

OPEN_STATUS
The full status, including the battery life, power source, and battery state, is shown.
OPEN_BATTERY
Only the battery life indicator is shown.

wpQueryDefStatusView - Remarks

When Advanced Power Management (APM) is enabled, this method returns the current default view of the power management object.

When APM is not enabled, this method still returns the status, but the status returned is not the default view because APM is disabled. The status returned is the status that will be the default the next time APM is enabled.

wpQueryDefStatusView - How to Override

This method should not be overridden.

wpQueryDefStatusView - Related Methods

Related Methods

- wpQueryPowerManagement
- wpSetDefStatusView
- wpSetPowerManagement

wpQueryDefStatusView - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpQueryDetailsData

wpQueryDetailsData - Syntax

This instance method is called to allow the object to query its current details data.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;        /*  Pointer to the object on which the method is being invoked. */
PVOID        *ppDetailsData;  /*  Pointer to detail data information. */
PULONG        pcp;            /*  Length of ppDetailsData buffer. */
ULONG         rc;             /*  Success indicator. */

rc = _wpQueryDetailsData(somSelf, ppDetailsData,
       pcp);
```

wpQueryDetailsData Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryDetailsData Parameter - ppDetailsData

ppDetailsData (PVOID *) - output
Pointer to detail data information.

wpQueryDetailsData Parameter - pcp

pcp (PULONG) - in/out
Length of *ppDetailsData* buffer.

If *ppDetailsData* is set to NULL, the actual size of *ppDetailsData* is returned in *pcp*.

wpQueryDetailsData Return Value - rc

rc (ULONG) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryDetailsData - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ppDetailsData (PVOID *) - output
Pointer to detail data information.

pcp (PULONG) - in/out
Length of *ppDetailsData* buffer.

If *ppDetailsData* is set to NULL, the actual size of *ppDetailsData* is returned in *pcp*.

rc (ULONG) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryDetailsData - Remarks

All objects which have information to display in *details* view must override this method. *ppDetailsData* is a pointer to the beginning of the buffer into which details data should be written. The override should write whatever data it is responsible for and then increment the pointer to the beginning of the area of the next class in the hierarchy (1 byte past the last field for which it is responsible). Note that *ppDetailsData* *must* be modified so that the subclasses write in the appropriate place.

The details data returned by a class must match the information returned in wpclsQueryDetailsInfo.

wpQueryDetailsData - Usage

This method can be called at any time in order to determine the current object details.

wpQueryDetailsData - How to Override

All objects which have information to display in *details* view must override this method.

Note that the parent method must always be called before writing the data and adjusting the pointer.

wpQueryDetailsData - Related Methods

Related Methods

- wpQueryDetailsData
- wpclsQueryDetails
- wpclsQueryDetailsInfo

wpQueryDetailsData - Example Code

This example returns the myfile specific data for the details view of this object and sets the pointer (*ppDetailsData) to the beginning of the buffer into which the data is written.

```
SOM_Scope ULONG SOMLINK myf_wpQueryDetailsData(MYFILE *somSelf,
                PVOID *ppDetailsData,
                PULONG pcp)

{
    PMYFILEDETAILS pMYFILEDetails;
    PBYTE       pSize;
    MYFILEData *somThis = MYFILEGetData(somSelf);
    MYFILEMethodDebug("MYFILE","myf_wpQueryDetailsData");

   parent_wpQueryDetailsData(somSelf,ppDetailsData, pcp);

   if (ppDetailsData)                                        /* query data */
   {
      pMYFILEDetails                 = (PMYFILEDETAILS) *ppDetailsData;
      pMYFILEDetails->pszLastWorker  = _queryLastWorker(somSelf);

      /* point to buffer location after our details data */
      *ppDetailsData = ((PBYTE) (*ppDetailsData)) + sizeof(*pMYFILEDetails);

   } else {
      *pcp += sizeof(*pMYFILEDetails);     /* caller is querying size of buffer */
   }

   return( TRUE );

}
```

wpQueryDetailsData - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryDisk

wpQueryDisk - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the disk object for the drive on which the file system object is located.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
WPFileSystem     *Disk;     /*  Disk object of the drive on which the file system is located. */

Disk = _wpQueryDisk(somSelf);
```

wpQueryDisk Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpQueryDisk Return Value - Disk

Disk (WPFileSystem *) - returns
Disk object of the drive on which the file system is located.

wpQueryDisk - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

Disk (WPFileSystem *) - returns
Disk object of the drive on which the file system is located.

wpQueryDisk - Topics

Select an item:
Syntax
Parameters
Returns

wpQueryDisplaySmallIcons

wpQueryDisplaySmallIcons - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method determines whether the small icons are displayed on the Toolbar.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL             fState;   /*  Flag indicating whether small icons are displayed. */

fState = _wpQueryDisplaySmallIcons(somSelf);
```

wpQueryDisplaySmallIcons Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpQueryDisplaySmallIcons Return Value - fState

fState (BOOL) - returns
Flag indicating whether small icons are displayed.

TRUE
Small icons are displayed.

FALSE
Normal-sized icons are displayed.

wpQueryDisplaySmallIcons - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

fState (BOOL) - returns
Flag indicating whether small icons are displayed.

TRUE
Small icons are displayed.
FALSE
Normal-sized icons are displayed.

wpQueryDisplaySmallIcons - How to Override

This method can be overridden to force a particular value.

wpQueryDisplaySmallIcons - Related Methods

Related Methods

- wpSetDisplaySmallIcons
- wpRefreshDrawer

wpQueryDisplaySmallIcons - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryDisplayText

wpQueryDisplayText - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method determines whether the text is displayed under icons on the Toolbar.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL             fState;   /*  Flag indicating whether text is displayed. */
```

```
fState = _wpQueryDisplayText(somSelf);
```

wpQueryDisplayText Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpQueryDisplayText Return Value - fState

fState (BOOL) - returns
Flag indicating whether text is displayed.

TRUE
The text is displayed.
FALSE
The text is not displayed.

wpQueryDisplayText - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

fState (BOOL) - returns
Flag indicating whether text is displayed.

TRUE
The text is displayed.
FALSE
The text is not displayed.

wpQueryDisplayText - How to Override

This method can be overridden to force a particular value.

wpQueryDisplayText - Related Methods

Related Methods

- wpQueryDisplayTextInDrawers
- wpRefreshDrawer
- wpSetDisplayText
- wpSetDisplayTextInDrawers

wpQueryDisplayText - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryDisplayTextInDrawers

wpQueryDisplayTextInDrawers - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method determines whether the text is displayed under icons in drawers.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL             fState;   /*  Flag indicating whether text is displayed. */

fState = _wpQueryDisplayTextInDrawers(somSelf);
```

wpQueryDisplayTextInDrawers Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpQueryDisplayTextInDrawers Return Value - fState

fState (BOOL) - returns

Flag indicating whether text is displayed.

TRUE
: The text is displayed.

FALSE
: The text is not displayed.

wpQueryDisplayTextInDrawers - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

fState (BOOL) - returns
Flag indicating whether text is displayed.

TRUE
: The text is displayed.

FALSE
: The text is not displayed.

wpQueryDisplayTextInDrawers - How to Override

This method can be overridden to force a particular value.

wpQueryDisplayTextInDrawers - Related Methods

Related Methods

- wpQueryDisplayText
- wpRefreshDrawer
- wpSetDisplayText
- wpSetDisplayTextInDrawers

wpQueryDisplayTextInDrawers - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryDisplayVertical

wpQueryDisplayVertical - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the orientation of the Toolbar.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL             fState;   /*  Flag indicating the orientation of the Toolbar. */

fState = _wpQueryDisplayVertical(somSelf);
```

wpQueryDisplayVertical Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpQueryDisplayVertical Return Value - fState

fState (BOOL) - returns
Flag indicating the orientation of the Toolbar.

TRUE
The Toolbar is displayed vertically.
FALSE
The Toolbar is displayed horizontally.

wpQueryDisplayVertical - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

fState (BOOL) - returns
Flag indicating the orientation of the Toolbar.

TRUE
The Toolbar is displayed vertically.
FALSE

The Toolbar is displayed horizontally.

wpQueryDisplayVertical - Remarks

Drawers are displayed in the opposite direction.

wpQueryDisplayVertical - How to Override

This method can be overridden to force a particular value.

wpQueryDisplayVertical - Related Methods

Related Methods

- wpSetDisplayVertical
- wpRefreshDrawer

wpQueryDisplayVertical - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpQueryDrawerHWND

wpQueryDrawerHWND - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the handle associated with the open Toolbar or drawer.

```
#define INCL_WINWORKPLACE
#include <os2.h>
```

```
WPLaunchPad     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
ULONG            ulDrawer;  /*  Toolbar or drawer being queried. */
HWND             hwnd;      /*  Handle of the frame window for the Toolbar or drawer. */

hwnd = _wpQueryDrawerHWND(somSelf, ulDrawer);
```

wpQueryDrawerHWND Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpQueryDrawerHWND Parameter - ulDrawer

ulDrawer (ULONG) - input
Toolbar or drawer being queried.

0
Toolbar.
>0
Drawer identifier.

wpQueryDrawerHWND Return Value - hwnd

hwnd (HWND) - returns
Handle of the frame window for the Toolbar or drawer.

A return value of NULL indicates that the Toolbar or the specified drawer is closed.

wpQueryDrawerHWND - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

ulDrawer (ULONG) - input
Toolbar or drawer being queried.

0
Toolbar.
>0
Drawer identifier.

hwnd (HWND) - returns
Handle of the frame window for the Toolbar or drawer.

A return value of NULL indicates that the Toolbar or the specified drawer is closed.

wpQueryDrawerHWND - How to Override

This method can be overridden to know when the slideup view handle is being requested.

wpQueryDrawerHWND - Related Methods

Related Methods

- wpRefreshDrawer
- wpSetDrawerHWND

wpQueryDrawerHWND - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryDriveLockStatus

wpQueryDriveLockStatus - Syntax

This method is specific to version 2.1, or higher, of the OS/2 operating system.

This instance method returns the lock status of a drive.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDisk     *somSelf;        /*  Pointer to the object on which the method is being invoked. */
PULONG      pulLockStatus;  /*  Pointer to the lock status of the drive. */
PULONG      pulLockCount;   /*  Number of locks against the drive. */
ULONG       rc;             /*  Return code from DosDevIOCtl category 8 Function 66. */

rc = _wpQueryDriveLockStatus(somSelf, pulLockStatus,
       pulLockCount);
```

wpQueryDriveLockStatus Parameter - somSelf

somSelf (WPDisk *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDisk.

wpQueryDriveLockStatus Parameter - pulLockStatus

pulLockStatus (PULONG) - input
Pointer to the lock status of the drive.

Possible values are described in the following list:

0
No media exists in the drive, and the drive is unlocked.

1
Media exists in the drive.

2
The drive is locked.

3
Media exists in the drive, and the drive is locked.

wpQueryDriveLockStatus Parameter - pulLockCount

pulLockCount (PULONG) - input
Number of locks against the drive.

wpQueryDriveLockStatus Return Value - rc

rc (ULONG) - returns
Return code from DosDevIOCtl category 8 Function 66.

A return value of NO_ERROR indicates the drive does not support lock and eject.

wpQueryDriveLockStatus - Parameters

somSelf (WPDisk *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDisk.

pulLockStatus (PULONG) - input
Pointer to the lock status of the drive.

Possible values are described in the following list:

0
No media exists in the drive, and the drive is unlocked.

1
Media exists in the drive.

2
The drive is locked.

3
Media exists in the drive, and the drive is locked.

pulLockCount (PULONG) - input
Number of locks against the drive.

rc (ULONG) - returns
Return code from DosDevIOCtl category 8 Function 66.

A return value of NO_ERROR indicates the drive does not support lock and eject.

wpQueryDriveLockStatus - Usage

This method can be called at any time to verify the lock status of a drive.

wpQueryDriveLockStatus - How to Override

This method should not be overridden.

wpQueryDriveLockStatus - Related Methods

Related Methods

- wpEjectDisk
- wpLockDrive

wpQueryDriveLockStatus - Example Code

```
ULONG   ulStatus    = 0;            /* Status returned from _wpLockDrive */
ULONG   ulLockCount = 0;            /* Number of times drive is locked   */

           /* Lock the target drive */

   ulStatus = _wpLockDrive( self, TRUE );

   if (ulStatus & DRIVE_LOCK_NO_SUPPORT) {
     somPrintf("Drive does not support lock/unlock.\n");
   } else {

          /* Query the target drive */
```

```
    rc = _wpQueryDriveLockStatus( self                /* Self pointer */
                                , &ulStatus           /* Lock status  */
                                , &ulLockCount );     /* Lock count   */

    if (rc == NO_ERROR) {

      somPrintf("_wpQueryDriveLockStatus reports:\n");

      if (ulStatus & DRIVE_LOCKED) {
        somLPrintf(1, "Drive is locked.\n");
        somLPrintf(1, "Lock count = %u\n", ulLockCount);
      }
    }

         /* Unlock the target drive */

    ulStatus = _wpLockDrive( self, FALSE );

  } /* endif */
```

wpQueryDriveLockStatus - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryEASize

wpQueryEASize - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the size of a file's extended attributes (EAs).

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
ULONG             ulEASize;  /*  Size of the file's extended attributes. */

ulEASize = _wpQueryEASize(somSelf);
```

wpQueryEASize Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpQueryEASize Return Value - ulEASize

ulEASize (ULONG) - returns
Size of the file's extended attributes.

wpQueryEASize - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

ulEASize (ULONG) - returns
Size of the file's extended attributes.

wpQueryEASize - How to Override

This method is generally not overridden.

wpQueryEASize - Related Methods

Related Methods

- wpQueryFileSize
- wpSetFileSizeInfo

wpQueryEASize - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryEASupport

wpQueryEASupport - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method is called to determine whether extended attributes are supported for the specified file system object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG             rc;       /*  Success indicator */

rc = _wpQueryEASupport(somSelf);
```

wpQueryEASupport Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpQueryEASupport Return Value - rc

rc (ULONG) - returns
Success indicator

NO_ERROR	Extended attributes supported.
ERROR_EAS_NOT_SUPPORTED	Extended attributes not supported.

wpQueryEASupport - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

rc (ULONG) - returns
Success indicator

NO_ERROR	Extended attributes supported.
ERROR_EAS_NOT_SUPPORTED	Extended attributes not supported.

wpQueryEASupport - Usage

This method can be called at any time in order to determine the whether or not the file system that contains the current object supports extended attributes.

wpQueryEASupport - How to Override

This method is generally not overridden.

wpQueryEASupport - Related Methods

Related Methods

- wpQueryEASize

wpQueryEASupport - Example Code

This example determines whether or not the file system for the current file system object supports extended attributes.

```
ULONG        ulEASupport;

ulEASupport = _wpQueryEASupport(somSelf);
/* ulEASupport is NO_ERROR or ERROR_EAS_NOT_SUPPORTED */
```

wpQueryEASupport - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryError

wpQueryError - Syntax

This instance method is called to retrieve the error identity of the last error condition.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
ULONG         ulerrorid;  /*  Error identities. */

ulerrorid = _wpQueryError(somSelf);
```

wpQueryError Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryError Return Value - ulerrorid

ulerrorid (ULONG) - returns
Error identities.

WPERR_PROTECTED_CLASS
Class is protected.
WPERR_INVALID_CLASS
Class is invalid.
WPERR_INVALID_SUPERCLASS
Superclass is invalid.
WPERR_NO_MEMORY
Out of memory.
WPERR_SEMAPHORE_ERROR
Semaphore error.
WPERR_BUFFER_TOO_SMALL
Buffer too small.
WPERR_CLSLOADMOD_FAILED
Unable to load the class library module.
WPERR_CLSPROCADDR_FAILED
Unable to find the class entry point.
WPERR_OBJWORD_LOCATION
Bad object word location.
WPERR_INVALID_OBJECT
Object pointer is invalid.
WPERR_MEMORY_CLEANUP
Cannot free memory.
WPERR_INVALID_MODULE
Invalid class library module.
NO_ERROR
No error conditions have been set.
WPERR_USER
Application-defined errors are above this value. This value can also include any of the base (ERROR_*) error codes.

WPERR_INVALID_OLDCLASS
Old class being replaced is invalid.
WPERR_INVALID_NEWCLASS
New replacement class is invalid.
WPERR_NOT_IMMEDIATE_CHILD
New class is not an immediate child of the old class.
WPERR_NOT_WORKPLACE_CLASS
Class is not a Workplace class.
WPERR_CANT_REPLACE_METACLS
Metaclass cannot be replaced.
WPERR_INI_FILE_WRITE
Cannot write to INI file.
WPERR_INVALID_FOLDER
Folder is invalid.
WPERR_BUFFER_OVERFLOW
Buffer overflow.
WPERR_OBJECT_NOT_FOUND
No objects were found.
WPERR_INVALID_HFIND
Handle for find operation is invalid.
WPERR_INVALID_COUNT
Count of objects to find is invalid.
WPERR_INVALID_BUFFER
Buffer is invalid.
WPERR_ALREADY_EXISTS
Object already exists.
WPERR_INVALID_FLAGS
Flags are invalid.
WPERR_INVALID_OBJECTID
Object id is invalid.
WPERR_INVALID_TARGET_OBJECT
Target object is invalid.

wpQueryError - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulerrorid (ULONG) - returns
Error identities.

WPERR_PROTECTED_CLASS
Class is protected.
WPERR_INVALID_CLASS
Class is invalid.
WPERR_INVALID_SUPERCLASS
Superclass is invalid.
WPERR_NO_MEMORY
Out of memory.
WPERR_SEMAPHORE_ERROR
Semaphore error.
WPERR_BUFFER_TOO_SMALL
Buffer too small.
WPERR_CLSLOADMOD_FAILED
Unable to load the class library module.
WPERR_CLSPROCADDR_FAILED
Unable to find the class entry point.
WPERR_OBJWORD_LOCATION
Bad object word location.
WPERR_INVALID_OBJECT
Object pointer is invalid.
WPERR_MEMORY_CLEANUP
Cannot free memory.
WPERR_INVALID_MODULE
Invalid class library module.
NO_ERROR
No error conditions have been set.

WPERR_USER
: Application-defined errors are above this value. This value can also include any of the base (ERROR_*) error codes.

WPERR_INVALID_OLDCLASS
: Old class being replaced is invalid.

WPERR_INVALID_NEWCLASS
: New replacement class is invalid.

WPERR_NOT_IMMEDIATE_CHILD
: New class is not an immediate child of the old class.

WPERR_NOT_WORKPLACE_CLASS
: Class is not a Workplace class.

WPERR_CANT_REPLACE_METACLS
: Metaclass cannot be replaced.

WPERR_INI_FILE_WRITE
: Cannot write to INI file.

WPERR_INVALID_FOLDER
: Folder is invalid.

WPERR_BUFFER_OVERFLOW
: Buffer overflow.

WPERR_OBJECT_NOT_FOUND
: No objects were found.

WPERR_INVALID_HFIND
: Handle for find operation is invalid.

WPERR_INVALID_COUNT
: Count of objects to find is invalid.

WPERR_INVALID_BUFFER
: Buffer is invalid.

WPERR_ALREADY_EXISTS
: Object already exists.

WPERR_INVALID_FLAGS
: Flags are invalid.

WPERR_INVALID_OBJECTID
: Object id is invalid.

WPERR_INVALID_TARGET_OBJECT
: Target object is invalid.

wpQueryError - Usage

This method can be called at any time in order to determine the identity of the last error that occurred. This method is typically called after calling a method that returned a failure.

wpQueryError - How to Override

This method is generally not overridden.

wpQueryError - Related Methods

Related Methods

- wpSetError

wpQueryError - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpQueryFilename

wpQueryFilename - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method copies the fully-qualified or unqualified file name to a buffer.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
PSZ               pszFilename;  /*  Pointer to the buffer containing the output file name. */
BOOL              fQualified;   /*  Qualification requested flag. */
PSZ               pszBuf;       /*  Pointer to the buffer containing the output filename. */

pszBuf = _wpQueryFilename(somSelf, pszFilename,
           fQualified);
```

wpQueryFilename Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpQueryFilename Parameter - pszFilename

pszFilename (PSZ) - output
Pointer to the buffer containing the output file name.

This buffer must be at least CCHMAXPATH characters in length.

wpQueryFilename Parameter - fQualified

fQualified (BOOL) - input
Qualification requested flag.

Possible values are as follows:

TRUE	Output file name is fully qualified.
FALSE	Output file name is unqualified.

wpQueryFilename Return Value - pszBuf

pszBuf (PSZ) - returns
Pointer to the buffer containing the output filename.

If the file exist, *pszFilename* is returned. If the file does not exist, NULL is returned.

wpQueryFilename - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

pszFilename (PSZ) - output
Pointer to the buffer containing the output file name.

This buffer must be at least CCHMAXPATH characters in length.

fQualified (BOOL) - input
Qualification requested flag.

Possible values are as follows:

TRUE	Output file name is fully qualified.
FALSE	Output file name is unqualified.

pszBuf (PSZ) - returns
Pointer to the buffer containing the output filename.

If the file exist, *pszFilename* is returned. If the file does not exist, NULL is returned.

wpQueryFilename - How to Override

This method should not be overridden.

wpQueryFilename - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Glossary

wpQueryFileSize

wpQueryFileSize - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the size of a given file.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG             rc;       /*  Size of the file. */

rc = _wpQueryFileSize(somSelf);
```

wpQueryFileSize Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpQueryFileSize Return Value - rc

rc (ULONG) - returns
Size of the file.

wpQueryFileSize - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

rc (ULONG) - returns
Size of the file.

wpQueryFileSize - How to Override

This method is generally not overridden.

wpQueryFileSize - Related Methods

Related Methods

- wpQueryEASize
- wpSetFileSizeInfo

wpQueryFileSize - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryFldrAttr

wpQueryFldrAttr - Syntax

This instance method is called to allow the folder to query its current view attributes for the WC_CONTAINER window used in each view window.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;           /*  Pointer to the object on which the method is being invoked. */
ULONG         ulView;            /*  Flag indicating the view to query. */
ULONG         ulViewAttributes;  /*  Flag containing current folder view attribute. */

ulViewAttributes = _wpQueryFldrAttr(somSelf,
                     ulView);
```

wpQueryFldrAttr Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryFldrAttr Parameter - ulView

ulView (ULONG) - input
Flag indicating the view to query.

OPEN_CONTENTS
Open contents view.
OPEN_DETAILS
Open details view.
OPEN_HELP
Display HelpPanel.
OPEN_TREE
Open tree view.

wpQueryFldrAttr Return Value - ulViewAttributes

ulViewAttributes (ULONG) - returns
Flag containing current folder view attribute.

These are the CV_* attributes defined by the Container Control Window. See CNRINFO for a detailed description.

wpQueryFldrAttr - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulView (ULONG) - input
Flag indicating the view to query.

OPEN_CONTENTS
Open contents view.
OPEN_DETAILS
Open details view.
OPEN_HELP
Display HelpPanel.
OPEN_TREE
Open tree view.

ulViewAttributes (ULONG) - returns

Flag containing current folder view attribute.

These are the CV_* attributes defined by the Container Control Window. See CNRINFO for a detailed description.

wpQueryFldrAttr - Usage

This method can be called at any time in order to determine the view attributes currently set.

wpQueryFldrAttr - How to Override

This method is generally not overridden.

wpQueryFldrAttr - Related Methods

Related Methods

- wpQueryFldrAttr
- wpQueryFldrFlags
- wpQueryFldrFont
- wpSetFldrAttr
- wpSetFldrFlags
- wpSetFldrFont

wpQueryFldrAttr - Example Code

```
SOMAny      *pfolder, *objDesktop, *ObjCopy;
BOOL        rc;
CHAR        FolderTitle[CCHMAXPATHCOMP] = "My Folder Title";
PSZ         pszFolderTitle = FolderTitle;
ULONG       ulFolderAttr;

  objDesktop = _wpclsQueryFolder(_WPFolder,"<WP_DESKTOP>",TRUE);

  if ((pfolder = _wpclsNew(_WPFolder,              /* Object class name  */
                           pszFolderTitle,         /* Title of object    */
                           "NODELETE=NO",          /* Setup string       */
                           objDesktop,             /* Target folder      */
                           TRUE)) != NULLHANDLE)   /* Never dormant      */

    {
    if ((ObjCopy = _wpCopyObject(self, pfolder, TRUE)) != NULLHANDLE) {

      ulFolderAttr = _wpQueryFldrAttr (pfolder, OPEN_CONTENTS);

      if (ulFolderAttr & CV_FLOW) {

           /* Handle view processing */

      } // endif FolderAttr

    } // endif wpclsNew
```

wpQueryFldrAttr - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryFldrBackground

wpQueryFldrBackground - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns the current values for the folder background.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;            /*  Pointer to the object on which the method is being invoked. */
PSZ          *ppszImageFileName;  /*  The name of the background image file (or NULL if none is defined). */
ULONG        *pulImageMode;       /*  The image mode. */
ULONG        *pulScaleFactor;     /*  The scaling factor for scaled image mode. */
ULONG        *pulBackgroundType;  /*  The background type. */
LONG         *plBackgroundColor;  /*  The RGB value of the background color. */

wpQueryFldrBackground(somSelf, ppszImageFileName,
       pulImageMode, pulScaleFactor, pulBackgroundType,
       plBackgroundColor);
```

wpQueryFldrBackground Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryFldrBackground Parameter - ppszImageFileName

ppszImageFileName (PSZ *) - output
The name of the background image file (or NULL if none is defined).

wpQueryFldrBackground Parameter - pulImageMode

pulImageMode (ULONG *) - output
The image mode.

BACKGROUND_NORMAL_IMAGE	Normal image.
BACKGROUND_TILED_IMAGE	Tiled image.
BACKGROUND_SCALED_IMAGE	Scaled image.

wpQueryFldrBackground Parameter - pulScaleFactor

pulScaleFactor (ULONG *) - output
The scaling factor for scaled image mode.

wpQueryFldrBackground Parameter - pulBackgroundType

pulBackgroundType (ULONG *) - output
The background type.

BACKGROUND_COLOR_ONLY	The background is color only (no image).
BACKGROUND_IMAGE	The background is an image.

wpQueryFldrBackground Parameter - plBackgroundColor

plBackgroundColor (LONG *) - output
The RGB value of the background color.

wpQueryFldrBackground - Return Value

There is no return value for this method.

The values for the folder background are returned in the parameters.

wpQueryFldrBackground - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ppszImageFileName (PSZ *) - output
The name of the background image file (or NULL if none is defined).

pulImageMode (ULONG *) - output
The image mode.

BACKGROUND_NORMAL_IMAGE	Normal image.
BACKGROUND_TILED_IMAGE	Tiled image.
BACKGROUND_SCALED_IMAGE	Scaled image.

pulScaleFactor (ULONG *) - output
The scaling factor for scaled image mode.

pulBackgroundType (ULONG *) - output
The background type.

BACKGROUND_COLOR_ONLY	The background is color only (no image).
BACKGROUND_IMAGE	The background is an image.

plBackgroundColor (LONG *) - output
The RGB value of the background color.

There is no return value for this method.

The values for the folder background are returned in the parameters.

wpQueryFldrBackground - Usage

This method can be called at any time in order to determine the background bitmap defintion for the folder.

wpQueryFldrBackground - How to Override

This method is generally not overridden.

wpQueryFldrBackground - Related Methods

Related Methods

- wpSetFldrBackground

wpQueryFldrBackground - Example Code

This examples lets you find the background definition for the current folder.

```
  PSZ           pszImageFile;
  ULONG         ulImageMode;
  ULONG         ulScaleFactor;
  ULONG         ulBackgroundType;
  LONG          lBackgroundColor;

  _wpQueryFldrBackground(somSelf,
                         &pszImageFileName,
                         &ulImageMode,
                         &ulScaleFactor,
                         &ulBackgroundType,
                         &lBackgroundColor);
  /* pszImageFileName contains the pointer to the name of the image file
   * ulImageMode is BACKGROUND_NORMAL_IMAGE,
   *                BACKGROUND_TILED_IMAGE,
   *                or BACKGROUND_SCALED_IMAGE
   * ulScaleFactor is a number between 1 and 20
   * ulBackgroundType is BACKGROUND_COLOR_ONLY or BACKGROUND_IMAGE
   * lBackgroundColor is an encoded RGB color value
   */
```

wpQueryFldrBackground - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryFldrDetailsClass

wpQueryFldrDetailsClass - Syntax

This instance method is called to determine which class of details are set for a folder.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder       *somSelf;  /*  Pointer to the object on which the method is being invoked. */
M_WPObject     *rc;       /*  Success indicator. */

rc = _wpQueryFldrDetailsClass(somSelf);
```

wpQueryFldrDetailsClass Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryFldrDetailsClass Return Value - rc

rc (M_WPObject *) - returns
Success indicator.

NULL	Error occurred.
Other	Pointer to the class object for which details are to be displayed.

wpQueryFldrDetailsClass - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

rc (M_WPObject *) - returns
Success indicator.

NULL	Error occurred.
Other	Pointer to the class object for which details are to be displayed.

wpQueryFldrDetailsClass - Remarks

Because folders can contain objects of different classes which can have different details, it is often necessary for the user to specify which class of details is to be displayed. The value set by this method is not used until a details view of the folder is opened. The wpSetFldrDetailsClass method can be called to set the current class of details to be displayed.

wpQueryFldrDetailsClass - Usage

This method can be called at any time in order to determine the current class of details to be displayed.

wpQueryFldrDetailsClass - How to Override

This method is generally not overridden.

wpQueryFldrDetailsClass - Related Methods

Related Methods

- wpSetFldrDetailsClass

wpQueryFldrDetailsClass - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpQueryFldrFlags

wpQueryFldrFlags - Syntax

This instance method is called to allow the folder to query its current flags.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
ULONG         ulReturn;  /*  Flag indicating the current folder state. */

ulReturn = _wpQueryFldrFlags(somSelf);
```

wpQueryFldrFlags Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryFldrFlags Return Value - ulReturn

ulReturn (ULONG) - returns
Flag indicating the current folder state.

FOI_POPULATEDWITHALL
This flag is set if the folder was populated with all its contents.

FOI_POPULATEDWITHFOLDERS
This flag is set if the folder was populated only with folders that it contains.

FOI_WORKAREA
This flag is set if the user sets the workarea property.

FOI_CHANGEFONT

FOI_WAMINIMIZE

FOI_WASTARTONRESTORE

FOI_NOREFRESHVIEWS

FOI_ASYNCREFRESHONOPEN

FOI_TREEPOPULATE

FOI_POPULATEINPROGRESS

FOI_REFRESHINPROGRESS

FOI_FIRSTPOPULATE

FOI_WAMCRINPROGRESS

FOI_CNRBKGNDOLDFORMAT

FOI_CHANGEICONBGNDCOLOR

FOI_CHANGEICONTEXTCOLOR

FOI_DELETEINPROGRESS

FOI_CHANGESHADOWTEXTCOLOR

wpQueryFldrFlags - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulReturn (ULONG) - returns
Flag indicating the current folder state.

FOI_POPULATEDWITHALL
This flag is set if the folder was populated with all its contents.

FOI_POPULATEDWITHFOLDERS
This flag is set if the folder was populated only with folders that it contains.

FOI_WORKAREA
This flag is set if the user sets the workarea property.

FOI_CHANGEFONT

FOI_WAMINIMIZE

FOI_WASTARTONRESTORE

FOI_NOREFRESHVIEWS

FOI_ASYNCREFRESHONOPEN

FOI_TREEPOPULATE

FOI_POPULATEINPROGRESS

FOI_REFRESHINPROGRESS

FOI_FIRSTPOPULATE

FOI_WAMCRINPROGRESS

FOI_CNRBKGNDOLDFORMAT

FOI_CHANGEICONBGNDCOLOR

FOI_CHANGEICONTEXTCOLOR

FOI_DELETEINPROGRESS

FOI_CHANGESHADOWTEXTCOLOR

wpQueryFldrFlags - Usage

This method is called at any time in order to determine a folder's flag state.

wpQueryFldrFlags - How to Override

This method is generally not overridden.

wpQueryFldrFlags - Related Methods

Related Methods

- wpQueryFldrAttr
- wpQueryFldrFlags
- wpQueryFldrFont
- wpSetFldrAttr
- wpSetFldrFlags
- wpSetFldrFont

wpQueryFldrFlags - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpQueryFldrFont

wpQueryFldrFont - Syntax

The wpQueryFldrFont instance method is called to allow the folder to query its current font.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
ULONG         ulView;       /*  Flag indicating the view to query. */
PSZ           pFontString;  /*  Pointer to font string for specified open view. */

pFontString = _wpQueryFldrFont(somSelf, ulView);
```

wpQueryFldrFont Parameter - somSelf

somSelf (WPFolder *) - input

Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryFldrFont Parameter - ulView

ulView (ULONG) - input
Flag indicating the view to query.

OPEN_CONTENTS
Open contents view.
OPEN_DETAILS
Open details view.
OPEN_TREE
Open tree view.

wpQueryFldrFont Return Value - pFontString

pFontString (PSZ) - returns
Pointer to font string for specified open view.

The font string is in the format of point size followed by a period followed by the face name.

For example, "10.Helvetica". (a presentation parameter string).

The maximum font string size is set to PPFONTSTRSIZE.

wpQueryFldrFont - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulView (ULONG) - input
Flag indicating the view to query.

OPEN_CONTENTS
Open contents view.
OPEN_DETAILS
Open details view.
OPEN_TREE
Open tree view.

pFontString (PSZ) - returns
Pointer to font string for specified open view.

The font string is in the format of point size followed by a period followed by the face name.

For example, "10.Helvetica". (a presentation parameter string).

The maximum font string size is set to PPFONTSTRSIZE.

wpQueryFldrFont - Remarks

There is only one font for each view. Concurrent views will all have the same font.

wpQueryFldrFont - Usage

This method can be called at any time in order to determine the current font for a view.

wpQueryFldrFont - How to Override

This method is generally not overridden.

wpQueryFldrFont - Related Methods

Related Methods

- wpQueryFldrAttr
- wpQueryFldrFlags
- wpQueryFldrFont
- wpSetFldrAttr
- wpSetFldrFlags
- wpSetFldrFont

wpQueryFldrFont - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpQueryFldrSort

wpQueryFldrSort - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns a pointer to the sort record for the specified open view and type.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
ULONG         ulView;       /*  Flag indicating the folder view. */
ULONG         ulType;       /*  Reserved - set to 0. */
PVOID         pSortRecord;  /*  Pointer to the SORTFASTINFO structure containing sort record information for the

pSortRecord = _wpQueryFldrSort(somSelf, ulView,
                ulType);
```

wpQueryFldrSort Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryFldrSort Parameter - ulView

ulView (ULONG) - input
Flag indicating the folder view.

Possible values are described in the following list:

OPEN_CONTENTS
Open content view.
OPEN_DETAILS
Open details view.

wpQueryFldrSort Parameter - ulType

ulType (ULONG) - input
Reserved - set to 0.

wpQueryFldrSort Return Value - pSortRecord

pSortRecord (PVOID) - returns
Pointer to the SORTFASTINFO structure containing sort record information for the specified view.

wpQueryFldrSort - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulView (ULONG) - input
Flag indicating the folder view.

Possible values are described in the following list:

OPEN_CONTENTS
Open content view.
OPEN_DETAILS
Open details view.

ulType (ULONG) - input
Reserved - set to 0.

pSortRecord (PVOID) - returns
Pointer to the SORTFASTINFO structure containing sort record information for the specified view.

wpQueryFldrSort - How to Override

This method should not be overridden.

wpQueryFldrSort - Related Methods

Related Methods

- wpQueryFldrSortClass
- wpSetFldrSort
- wpSetFldrSortClass

wpQueryFldrSort - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryFldrSortClass

wpQueryFldrSortClass - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the sort class object whose details are currently being used as sort criteria for a given folder.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder       *somSelf;  /*  Pointer to the object on which the method is being invoked. */
M_WPObject     *Class;    /*  Pointer to the class object. */

Class = _wpQueryFldrSortClass(somSelf);
```

wpQueryFldrSortClass Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryFldrSortClass Return Value - Class

Class (M_WPObject *) - returns
Pointer to the class object.

wpQueryFldrSortClass - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

Class (M_WPObject *) - returns
Pointer to the class object.

wpQueryFldrSortClass - Remarks

The "sort class" for a particular folder determines what class's sort details appear as active sort criteria for the folder's Settings page and sort

submenu. The default sort class for all folders is WPFileSystem. The sort class can be manually changed via the folder's Settings page or can be programmatically changed via wpSetFldrSortClass.

wpQueryFldrSortClass - How to Override

This method is generally not overridden.

wpQueryFldrSortClass - Related Methods

Related Methods

- wpQueryFldrSort
- wpSetFldrSort
- wpSetFldrSortClass

wpQueryFldrSortClass - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpQueryFloatOnTop

wpQueryFloatOnTop - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method determines whether the Toolbar floats on top of all other windows.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL             fState;   /*  Flag indicating whether the Toolbar floats on top of all other windows. */

fState = _wpQueryFloatOnTop(somSelf);
```

wpQueryFloatOnTop Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpQueryFloatOnTop Return Value - fState

fState (BOOL) - returns
Flag indicating whether the Toolbar floats on top of all other windows.

TRUE
The Toolbar floats on top.
FALSE
The Toolbar does not float on top.

wpQueryFloatOnTop - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

fState (BOOL) - returns
Flag indicating whether the Toolbar floats on top of all other windows.

TRUE
The Toolbar floats on top.
FALSE
The Toolbar does not float on top.

wpQueryFloatOnTop - How to Override

This method can be overridden to force a particular value.

wpQueryFloatOnTop - Related Methods

Related Methods

- wpSetFloatOnTop
- wpRefreshDrawer

wpQueryFloatOnTop - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryFolder

wpQueryFolder - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the folder that this object resides in.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
WPObject     *Object;   /*  Pointer to the container object that contains the given object. */

Object = _wpQueryFolder(somSelf);
```

wpQueryFolder Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryFolder Return Value - Object

Object (WPObject *) - returns
Pointer to the container object that contains the given object.

All objects are contained by a folder except for a drive object (root directory folder) for which this method returns NULL.

wpQueryFolder - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

Object (WPObject *) - returns
Pointer to the container object that contains the given object.

All objects are contained by a folder except for a drive object (root directory folder) for which this method returns NULL.

wpQueryFolder - Usage

This method can be called at any time to query the folder which contains the current object.

wpQueryFolder - How to Override

This method is generally not overridden.

wpQueryFolder - Related Methods

Related Methods

- wpclsQueryFolder

wpQueryFolder - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpQueryHandle

wpQueryHandle - Syntax

This instance method returns a unique, persistent object handle for the given object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
HOBJECT       hObject;  /*  Persistent handle for the given object. */

hObject = _wpQueryHandle(somSelf);
```

wpQueryHandle Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryHandle Return Value - hObject

hObject (HOBJECT) - returns
Persistent handle for the given object.

The high word of this handle is filled out by WPObject to identify the base class. The low word is filled out by the base class to allow it to locate the object.

wpQueryHandle - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hObject (HOBJECT) - returns
Persistent handle for the given object.

The high word of this handle is filled out by WPObject to identify the base class. The low word is filled out by the base class to allow it to locate the object.

wpQueryHandle - Remarks

The object handle returned from this method is the same handle that can either be used for the WinCreateObject, WinSetObjectData or WinDestroyObject. The handle is unique on a given machine so the returned object handle can be passed to other processes or stored for later use (even across IPLs of the system). For more information about WinCreateObject, WinSetObjectData, and WinDestroyObject, see the *Presentation Manager Programming Reference* .

Note: It is recommended that wpQueryHandle be used sparingly on file-based object classes because the operating system keeps track of the current location of every file object that has been allocated an object handle. Performance could be adversely affected if object handles are obtained for every file-system object that was ever awakened.

Using HOBJECT for .INI files or files in which an application uses a rename/save/delete sequence is not supported.

wpQueryHandle - Usage

This method can be called at any time to get a handle that is both persistent across IPLs and completely unique on the given machine.

wpQueryHandle - How to Override

This method is generally not overridden.

wpQueryHandle - Related Methods

Related Methods

- wpclsQueryObject

wpQueryHandle - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpQueryHandleFromContents

wpQueryHandleFromContents - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns a handle to a known system data type based on a file's contents. This is an abstract method; subclasses that describe files for which a system handle type exist (such as WPIcon or WPBitmap) should subclass this method.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDataFile     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
LHANDLE         handle;   /*  Handle to the data type object (or NULLHANDLE if call is unsuccessful). */

handle = _wpQueryHandleFromContents(somSelf);
```

wpQueryHandleFromContents Parameter - somSelf

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

wpQueryHandleFromContents Return Value - handle

handle (LHANDLE) - returns
Handle to the data type object (or NULLHANDLE if call is unsuccessful).

wpQueryHandleFromContents - Parameters

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

handle (LHANDLE) - returns
Handle to the data type object (or NULLHANDLE if call is unsuccessful).

wpQueryHandleFromContents - Remarks

The handle returned by this method is data type specific. For example, an Image File object will return a handle to a bitmap (that is, an HBITMAP), while an Icon or Pointer might return a handle to a pointer (such as an HPOINTER).

wpQueryHandleFromContents - Usage

This method can be called at any time to obtain a handle that represents the data in the data file corresponding to this data file object.

wpQueryHandleFromContents - How to Override

This method should be overridden in any subclass of WPDataFile that can represent its data with a handle.

wpQueryHandleFromContents - Related Methods

Related Methods

- wpQueryPointerFromContents
- wpSetContentsFromHandle
- wpSetContentsFromPointer

wpQueryHandleFromContents - Example Code

This example shows how the Image File class returns the handle to a bitmap representing the data stored in the data file. Notice how this method calls the wpQueryBitmapHandle method so all subclasses of WPImageFile can provide this function without adding any extra code.

```
SOM_Scope LHANDLE SOMLINK img_wpQueryHandleFromContents(WPImageFile
*somSelf)
{
   HBITMAP  hBitmap = NULLHANDLE;

   _wpQueryBitmapHandle(somSelf,&hBitmap,NULL,0,0,0,RGB_BLACK,NULL);
   return hBitmap;
}
```

wpQueryHandleFromContents - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryHideLaunchPadFrameCtls

wpQueryHideLaunchPadFrameCtls - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method determines whether the system menu and the title bar for the Toolbar are hidden.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL             fState;   /*  Flag indicating whether Frame controls are hidden. */

fState = _wpQueryHideLaunchPadFrameCtls(somSelf);
```

wpQueryHideLaunchPadFrameCtls Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpQueryHideLaunchPadFrameCtls Return Value - fState

fState (BOOL) - returns
Flag indicating whether Frame controls are hidden.

TRUE
The frame controls are hidden.
FALSE
The frame controls are not hidden.

wpQueryHideLaunchPadFrameCtls - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

fState (BOOL) - returns
Flag indicating whether Frame controls are hidden.

TRUE
The frame controls are hidden.
FALSE
The frame controls are not hidden.

wpQueryHideLaunchPadFrameCtls - How to Override

This method can be overridden to force a particular value.

wpQueryHideLaunchPadFrameCtls - Related Methods

Related Methods

- wpRefreshDrawer
- wpSetHideLaunchPadFrameCtls

wpQueryHideLaunchPadFrameCtls - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryIcon

wpQueryIcon - Syntax

This instance method is called to allow the object to query its current icon.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
HPOINTER      hptrQryIcon;  /*  Success indicator. */

hptrQryIcon = _wpQueryIcon(somSelf);
```

wpQueryIcon Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryIcon Return Value - hptrQryIcon

hptrQryIcon (HPOINTER) - returns
Success indicator.

NULLHANDLE
Error occurred
Other
Handle to an icon.

wpQueryIcon - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hptrQryIcon (HPOINTER) - returns
Success indicator.

NULLHANDLE
Error occurred
Other
Handle to an icon.

wpQueryIcon - Usage

This method can be called at any time in order to get the handle to the current icon for this object.

wpQueryIcon - How to Override

This method is generally not overridden. The default icon for a class is typically set from an override of wpclsQueryIcon and the instance's icon may be altered with wpSetIcon.

wpQueryIcon - Related Methods

Related Methods

- wpclsQueryIcon
- wpQueryIconData
- wpSetIcon
- wpSetIconData
- wpQueryIcon

wpQueryIcon - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpQueryIconData

wpQueryIconData - Syntax

This instance method is called to allow the object to query the data to be used for its current icon.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
PICONINFO     pIconInfo;   /*  Pointer to a buffer. */
ULONG         ulReturn;    /*  Buffer size or number of bytes required to hold the output data. */

ulReturn = _wpQueryIconData(somSelf, pIconInfo);
```

wpQueryIconData Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryIconData Parameter - pIconInfo

pIconInfo (PICONINFO) - output
Pointer to a buffer.

Pointer to the buffer large enough to contain an ICONINFO data structure followed by the data needed to describe the icon.

If *pIconInfo* is NULL, the return value is the required size of the buffer.

wpQueryIconData Return Value - ulReturn

ulReturn (ULONG) - returns
Buffer size or number of bytes required to hold the output data.

Depending of the contents of the *pIconInfo* parameter, *ulReturn* contains one of the following:

pIconInfo	*ulReturn*
NULL	Number of bytes required to hold the output data.
Not NULL	Number of bytes written into the buffer.

If *ulReturn* is 0, an error occurred.

wpQueryIconData - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pIconInfo (PICONINFO) - output
Pointer to a buffer.

Pointer to the buffer large enough to contain an ICONINFO data structure followed by the data needed to describe the icon.

If *pIconInfo* is NULL, the return value is the required size of the buffer.

ulReturn (ULONG) - returns
Buffer size or number of bytes required to hold the output data.

Depending of the contents of the *pIconInfo* parameter, *ulReturn* contains one of the following:

pIconInfo	*ulReturn*
NULL	Number of bytes required to hold the output data.
Not NULL	Number of bytes written into the buffer.

If *ulReturn* is 0, an error occurred.

wpQueryIconData - Remarks

If NULL is passed for the *pIconInfo* parameter, the caller is asking for the size of the ICONINFO buffer needed for this instance (usually for memory allocation purposes). Otherwise, the *pIconInfo* parameter can always be assumed to be large enough to accommodate the ICONINFO buffer and the variable data.

Note that the ICONINFO structure allows you to specify the default icon in three different ways:

- Block of binary data
- Icon file name
- Module name and resource identifier

However, only one mechanism need be supported. For example, a caller cannot request one of the three formats by prefilling the ICONINFO structure.

wpQueryIconData - Usage

This method is called at any time in order to query the data for the current icon for this object.

wpQueryIconData - How to Override

Workplace classes that wish to have a unique icon for each instance must override this method and fill out the appropriate fields within the ICONINFO structure. In addition, the correct size for the ICONINFO must always be returned.

This example shows how to override wpQueryIconData by returning information about the icon associated with the instance if *pIconInfo* is not NULL.

```
SOM_Scope ULONG SOMLINK Calendar_wpQueryIconData(_Calendar *somSelf,
                                                 PICONINFO pIconInfo)
{
  _CalendarData *somThis = _CalendarGetData(somSelf);
  _CalendarMethodDebug("_Calendar","myf_wpQueryIconData");

  if (pIconInfo)
  {
    pIconInfo->fFormat = ICON_RESOURCE;
    pIconInfo->hmod    = _clsQueryModuleHandle(_somGetClass(somSelf));
    pIconInfo->resid   = ID_ICON;
  } /* endif */
  return (sizeof(ICONINFO));
}
```

wpQueryIconData - Related Methods

Related Methods

- wpQueryIcon
- wpSetIcon
- wpSetIconData
- wpclsQueryIcon
- wpclsQueryIconData

wpQueryIconData - Example Code

This example code shows how to call wpQueryIconData.

```
cbIconSize = _wpQueryIconData(Object,NULL);
pIconSize  = (PICONINFO)_wpAllocMem(Object,cbIconSize,NULL);
if (pIconInfo != NULL)
{
  _wpQueryIconData(Object,pIconInfo);
}
```

wpQueryIconData - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryIconPosition

wpQueryIconPosition - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method finds the saved icon postion for this record.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
PSZ           pszIdentity;  /*  Object identity string. */
PPOINTL       pptl;         /*  A pointer to a POINTL structure containing the X- and Y-coordinates of the icon.
PULONG        pIndex;       /*  Reserved.  Set to 0. */
BOOL          rc;           /*  Success indicator */

rc = _wpQueryIconPosition(somSelf, pszIdentity,
       pptl, pIndex);
```

wpQueryIconPosition Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryIconPosition Parameter - pszIdentity

pszIdentity (PSZ) - input
Object identity string.

wpQueryIconPosition Parameter - pptl

pptl (PPOINTL) - input
A pointer to a POINTL structure containing the X- and Y-coordinates of the icon.

wpQueryIconPosition Parameter - pIndex

pIndex (PULONG) - input
Reserved. Set to 0.

wpQueryIconPosition Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryIconPosition - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

pszIdentity (PSZ) - input
Object identity string.

pptl (PPOINTL) - input
A pointer to a POINTL structure containing the X- and Y-coordinates of the icon.

pIndex (PULONG) - input
Reserved. Set to 0.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryIconPosition - Remarks

This method functions only when the folder is opened. To conserve memory, the ICONPOS data is thrown away after the initial records are

placed.

This method is only valid in between calls to wpInitIconPosData and wpFreeIconPosData (that is, when the icon position information is in memory).

wpQueryIconPosition - Usage

You can call the wpQueryIconPosition method to obtain the position of the icon for an object within an open Icon View of an folder. This method can be called only between calls to the wpInitIconPosData and wpFreeIconPosData methods.

wpQueryIconPosition - How to Override

This method is not generally overridden.

wpQueryIconPosition - Related Methods

Related Methods

- wpFreeIconPosData
- wpIdentify
- wpInitIconPosData
- wpStoreIconPosData

wpQueryIconPosition - Example Code

This example shows how you can use the wpInitIconPosData, wpIdentify, wpQueryIconPosition, and wpFreeIconPosData methods to find the icon position information for all objects within a folder. In this example, somSelf contains the pointer to the folder being examined.

```
BOOL        bSem;
WPObject    *Object;
CHAR        szIdentity[CCHMAXSTRING];
POINTL      ptl;
ULONG       ulIndex;

/* Retrieve the icon position information for all objects within
 * the folder    */
_wpInitIconPosData(somSelf);

/* Find the every object within the folder   */
bSem = !_wpRequestObjectMutexSem(somSelf,SEM_INDEFINITE_WAIT);
for (Object = _wpQueryContent(somSelf,NULL,QC_FIRST);
     Object;
     Object = _wpQueryContent(somSelf,Object,QC_NEXT));
{

   /* For each object in the folder, find the position of its icon
    * in an Icon View of the folder   */
   if (_wpIdentify(Object,szIdentity))
   {
      if (_wpQueryIconPosition(somSelf,szIdentity,&ptl,&ulIndex))
      {
         /***** ptl contains the position of the icon for this object *****/
      }
   }
}
if (bSem)
```

```
    {
       _wpReleaseObjectMutexSem(somSelf);
    }

    /* Free the icon position data structure    */
    _wpFreeIconPosData(somSelf);
```

wpQueryIconPosition - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryIconTextBackgroundColor

wpQueryIconTextBackgroundColor - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns the current color being used for the icon text background.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         rc;       /*  The RGB color value of the current color being used for the icon text background. */

rc = _wpQueryIconTextBackgroundColor(somSelf);
```

wpQueryIconTextBackgroundColor Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryIconTextBackgroundColor Return Value - rc

rc (ULONG) - returns
The RGB color value of the current color being used for the icon text background.

The RGB value is presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

wpQueryIconTextBackgroundColor - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

rc (ULONG) - returns
The RGB color value of the current color being used for the icon text background.

The RGB value is presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

wpQueryIconTextBackgroundColor - Usage

This method can be called at any time in order to determine the background text color used for normal text in Icon, Tree, or Details View of the folder.

wpQueryIconTextBackgroundColor - How to Override

This method is generally not overridden.

wpQueryIconTextBackgroundColor - Related Methods

Related Methods

- wpQueryIconTextColor
- wpQueryIconTextVisibility
- wpQueryShadowTextColor
- wpSetIconTextBackgroundColor
- wpSetIconTextColor
- wpSetIconTextVisibility
- wpSetShadowTextColor

wpQueryIconTextBackgroundColor - Example Code

This example lets you find the icon text background color for the current folder.

```
  ULONG        ulBackgroundColor;

  ulBackgroundColor = _wpQueryIconTextBackgroundColor(somSelf);
  /* ulBackgroundColor is an encoded RGB color value */
```

wpQueryIconTextBackgroundColor - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryIconTextColor

wpQueryIconTextColor - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns the current color being used for the icon text in the specified view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         ulView;   /*  Specifies which view to query. */
ULONG         rc;       /*  The RGB color value of the current color being used for the icon text background. */

rc = _wpQueryIconTextColor(somSelf, ulView);
```

wpQueryIconTextColor Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryIconTextColor Parameter - ulView

ulView (ULONG) - input
Specifies which view to query.

OPEN_CONTENTS
Open contents view.
OPEN_DEFAULT
Open default view (same as double-click).
OPEN_DETAILS
Open details view.
OPEN_HELP
Display HelpPanel.
OPEN_RUNNING
Execute object.
OPEN_SETTINGS
Open Settings notebook.
OPEN_TREE
Open tree view.
OPEN_USER
Class-specific views have a greater value than this.

wpQueryIconTextColor Return Value - rc

rc (ULONG) - returns
The RGB color value of the current color being used for the icon text background.

The RGB value is presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

wpQueryIconTextColor - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulView (ULONG) - input
Specifies which view to query.

OPEN_CONTENTS
Open contents view.
OPEN_DEFAULT
Open default view (same as double-click).
OPEN_DETAILS
Open details view.
OPEN_HELP
Display HelpPanel.
OPEN_RUNNING
Execute object.
OPEN_SETTINGS
Open Settings notebook.
OPEN_TREE
Open tree view.
OPEN_USER
Class-specific views have a greater value than this.

rc (ULONG) - returns

The RGB color value of the current color being used for the icon text background.

The RGB value is presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

wpQueryIconTextColor - Usage

This method can be called at any time in order to determine the color used to display the title of a normal icon in a view of the folder.

wpQueryIconTextColor - How to Override

This method is generally not overridden.

wpQueryIconTextColor - Related Methods

Related Methods

- wpQueryIconTextBackgroundColor
- wpQueryIconTextVisibility
- wpQueryShadowTextColor
- wpSetIconTextBackgroundColor
- wpSetIconTextColor
- wpSetIconTextVisibility
- wpSetShadowTextColor

wpQueryIconTextColor - Example Code

This example lets you find the icon text color for Icon View of the current folder.

```
ULONG        ulTextColor;

ulTextColor = _wpQueryIconTextColor(somSelf,OPEN_CONTENTS);
/* ulBackgroundColor is an encoded RGB color value */
```

wpQueryIconTextColor - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryIconTextVisibility

wpQueryIconTextVisibility - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns the current state of icon text visibility for the specified view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         ulView;   /*  Specifies which view to query. */
ULONG         rc;       /*  The current state of icon text visibility. */

rc = _wpQueryIconTextVisibility(somSelf, ulView);
```

wpQueryIconTextVisibility Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryIconTextVisibility Parameter - ulView

ulView (ULONG) - input
Specifies which view to query.

OPEN_CONTENTS
Opens the Contents view.
OPEN_DEFAULT
Opens the default view (same as double-clicking).
OPEN_DETAILS
Opens the Details view.
OPEN_HELP
Displays the Help panel.
OPEN_RUNNING
Executes the object.
OPEN_SETTINGS
Opens the Settings notebook.
OPEN_TREE
Opens the Tree view.
OPEN_USER
Class-specific views have a greater value than this.

wpQueryIconTextVisibility Return Value - rc

rc (ULONG) - returns
The current state of icon text visibility.

Byte 1	Has the following values:	
	0x00	Icon text is not visible.
	0x01	Icon text is visible (ICON_TEXT_VISIBLE).
Byte 4	Has the following values:	
	0x00	Instance setting
	0x40	Global setting (USE_GLOBAL_VISIBILITY).

wpQueryIconTextVisibility - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulView (ULONG) - input
Specifies which view to query.

OPEN_CONTENTS
Opens the Contents view.
OPEN_DEFAULT
Opens the default view (same as double-clicking).
OPEN_DETAILS
Opens the Details view.
OPEN_HELP
Displays the Help panel.
OPEN_RUNNING
Executes the object.
OPEN_SETTINGS
Opens the Settings notebook.
OPEN_TREE
Opens the Tree view.
OPEN_USER
Class-specific views have a greater value than this.

rc (ULONG) - returns
The current state of icon text visibility.

Byte 1	Has the following values:	
	0x00	Icon text is not visible.
	0x01	Icon text is visible (ICON_TEXT_VISIBLE).
Byte 4	Has the following values:	
	0x00	Instance setting
	0x40	Global setting (USE_GLOBAL_VISIBILITY).

wpQueryIconTextVisibility - Usage

This method can be called at any time in order to determine whether or not the titles are displayed in a view of the folder.

wpQueryIconTextVisibility - How to Override

This method is generally not overridden.

wpQueryIconTextVisibility - Related Methods

Related Methods

- wpSetIconTextVisibility

wpQueryIconTextVisibility - Example Code

This example determines whether or not the titles are displayed in the Icon View of the current folder.

```
ULONG       ulTextVisibility;

ulTextVisibility = _wpQueryIconTextVisibility(somSelf,OPEN_CONTENTS);
/* ulTextVisibility can contain the following flag values:
 *     USE_GLOBAL_VISIBILITY = the folder is using the global system value
 *     ICON_TEXT_VISIBLE     = titles are displayed in Icon View of the folder
 */
```

wpQueryIconTextVisibility - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryIconViewPos

wpQueryIconViewPos - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the initial icon view position of a given folder.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
CHAR         *IconViewPos[4];  /*  Pointer to the string containing the initial icon view position x, y, cx, and

IconViewPos[4] = _wpQueryIconViewPos(
                   somSelf);
```

wpQueryIconViewPos Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryIconViewPos Return Value - IconViewPos[4]

IconViewPos[4] (CHAR *) - returns
Pointer to the string containing the initial icon view position *x*, *y*, *cx*, and *cy*.

wpQueryIconViewPos - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

IconViewPos[4] (CHAR *) - returns
Pointer to the string containing the initial icon view position *x*, *y*, *cx*, and *cy*.

wpQueryIconViewPos - How to Override

This method is generally not overridden.

wpQueryIconViewPos - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Glossary

wpQueryJobFile

wpQueryJobFile - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the spool file name for the local print jobs.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPJob     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PSZ        pBuf;     /*  Buffer in which the complete path to the spool file is returned. */
PULONG     pcbBuf;   /*  Pointer to the length, in bytes, of the pBuf buffer. */
BOOL       rc;       /*  Success indicator. */

rc = _wpQueryJobFile(somSelf, pBuf, pcbBuf);
```

wpQueryJobFile Parameter - somSelf

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

wpQueryJobFile Parameter - pBuf

pBuf (PSZ) - output
Buffer in which the complete path to the spool file is returned.

A return value of NULL indicates that *pcbBuf* is too small to hold the network ID.

wpQueryJobFile Parameter - pcbBuf

pcbBuf (PULONG) - in/out
Pointer to the length, in bytes, of the *pBuf* buffer.

If either *pBuf* is NULL or this call fails, the length of the required buffer plus one is returned in this parameter.

wpQueryJobFile Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpQueryJobFile - Parameters

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

pBuf (PSZ) - output
Buffer in which the complete path to the spool file is returned.

A return value of NULL indicates that *pcbBuf* is too small to hold the network ID.

pcbBuf (PULONG) - in/out
Pointer to the length, in bytes, of the *pBuf* buffer.

If either *pBuf* is NULL or this call fails, the length of the required buffer plus one is returned in this parameter.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpQueryJobFile - Usage

This method can be called at any time.

wpQueryJobFile - How to Override

This method is generally not overridden.

wpQueryJobFile - Related Methods

Related Methods

- wpQueryJobId
- wpQueryJobType

wpQueryJobFile - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpQueryJobId

wpQueryJobId - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the job ID of the current job.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPJob     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG      ulJobId;  /*  Job ID of the current object. */

ulJobId = _wpQueryJobId(somSelf);
```

wpQueryJobId Parameter - somSelf

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

wpQueryJobId Return Value - ulJobId

ulJobId (ULONG) - returns
Job ID of the current object.

wpQueryJobId - Parameters

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

ulJobId (ULONG) - returns
Job ID of the current object.

wpQueryJobId - Usage

This method can be called at any time.

wpQueryJobId - How to Override

This method is generally not overridden.

wpQueryJobId - Related Methods

Related Methods

- wpQueryJobFile
- wpQueryJobType

wpQueryJobId - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpQueryJobType

wpQueryJobType - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the data type of the given job.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPJob     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PSZ        pBuf;     /*  Buffer in which the data type of the spooled job is returned. */
PULONG     pcbBuf;   /*  Pointer to the length, in bytes, of the pBuf buffer. */
BOOL       rc;       /*  Success indicator. */

rc = _wpQueryJobType(somSelf, pBuf, pcbBuf);
```

wpQueryJobType Parameter - somSelf

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

wpQueryJobType Parameter - pBuf

pBuf (PSZ) - output
Buffer in which the data type of the spooled job is returned.

A return value of NULL indicates that *pcbBuf* is too small to hold the datatype string.

Example data types are "PM_Q_STD" or "PM_Q_RAW."

wpQueryJobType Parameter - pcbBuf

pcbBuf (PULONG) - in/out
Pointer to the length, in bytes, of the *pBuf* buffer.

If either *pBuf* is NULL or this call fails, the length of the required buffer plus one is returned in this parameter.

wpQueryJobType Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpQueryJobType - Parameters

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

pBuf (PSZ) - output
Buffer in which the data type of the spooled job is returned.

A return value of NULL indicates that *pcbBuf* is too small to hold the datatype string.

Example data types are "PM_Q_STD" or "PM_Q_RAW."

pcbBuf (PULONG) - in/out
Pointer to the length, in bytes, of the *pBuf* buffer.

If either *pBuf* is NULL or this call fails, the length of the required buffer plus one is returned in this parameter.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpQueryJobType - How to Override

This method is generally not overridden.

wpQueryJobType - Related Methods

Related Methods

- wpQueryJobId
- wpQueryJobFile

wpQueryJobType - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryLastAccess

wpQueryLastAccess - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the last date and time a given file was accessed.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
FDATE            *fdate;    /*  Pointer to the date structure containing the last date the file was accessed. */
FTIME            *ftime;    /*  Pointer to the time structure containing the last time the file was accessed. */
ULONG             rc;       /*  Success indicator. */

rc = _wpQueryLastAccess(somSelf, fdate, ftime);
```

wpQueryLastAccess Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpQueryLastAccess Parameter - fdate

fdate (FDATE *) - output
Pointer to the date structure containing the last date the file was accessed.

wpQueryLastAccess Parameter - ftime

ftime (FTIME *) - output
Pointer to the time structure containing the last time the file was accessed.

wpQueryLastAccess Return Value - rc

rc (ULONG) - returns
Success indicator.

0
No error.
other
Error occurred.

wpQueryLastAccess - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

fdate (FDATE *) - output
Pointer to the date structure containing the last date the file was accessed.

ftime (FTIME *) - output
Pointer to the time structure containing the last time the file was accessed.

rc (ULONG) - returns
Success indicator.

0
No error.
other
Error occurred.

wpQueryLastAccess - How to Override

This method is generally not overridden.

wpQueryLastAccess - Related Methods

Related Methods

- wpQueryCreation
- wpQueryLastWrite

wpQueryLastAccess - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryLastWrite

wpQueryLastWrite - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the last date and time a given file was written to.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
FDATE            *fdate;    /*  Pointer to the date structure containing the last date the file was written to.
FTIME            *ftime;    /*  Pointer to the time structure containing the last time the file was written to.
ULONG             rc;       /*  Success indicator. */

rc = _wpQueryLastWrite(somSelf, fdate, ftime);
```

wpQueryLastWrite Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpQueryLastWrite Parameter - fdate

fdate (FDATE *) - output
Pointer to the date structure containing the last date the file was written to.

wpQueryLastWrite Parameter - ftime

ftime (FTIME *) - output
Pointer to the time structure containing the last time the file was written to.

wpQueryLastWrite Return Value - rc

rc (ULONG) - returns
Success indicator.

0
No error.
other
Error occurred.

wpQueryLastWrite - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

fdate (FDATE *) - output
Pointer to the date structure containing the last date the file was written to.

ftime (FTIME *) - output
Pointer to the time structure containing the last time the file was written to.

rc (ULONG) - returns
Success indicator.

0
No error.
other
Error occurred.

wpQueryLastWrite - How to Override

This method is generally not overridden.

wpQueryLastWrite - Related Methods

Related Methods

- wpQueryCreation
- wpQueryLastAccess

wpQueryLastWrite - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryLocalAlias

wpQueryLocalAlias - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the local alias of the remote printer object. The local alias is the local print queue as defined in the OS/2 print spooler.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPRPrinter     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PSZ             pBuf;     /*  Buffer in which the local print queue name is returned. */
PULONG          pcbBuf;   /*  Pointer to the length, in bytes, of the pBuf buffer. */
BOOL            rc;       /*  Success indicator. */

rc = _wpQueryLocalAlias(somSelf, pBuf, pcbBuf);
```

wpQueryLocalAlias Parameter - somSelf

somSelf (WPRPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPRPrinter.

wpQueryLocalAlias Parameter - pBuf

pBuf (PSZ) - output
Buffer in which the local print queue name is returned.

wpQueryLocalAlias Parameter - pcbBuf

pcbBuf (PULONG) - in/out
Pointer to the length, in bytes, of the *pBuf* buffer.

If *pBuf* points to zero, or this call fails, the length of the required buffer plus one is returned in this parameter.

wpQueryLocalAlias Return Value - rc

rc (BOOL) - returns
Success indicator.

If the buffer is too small, *pcbBuf* will contain the required size of the buffer. If there does not exist a local alias for this remoter printer object, *pcbBuf* will contain zero.

TRUE
Successful completion.
FALSE
Error occurred.

wpQueryLocalAlias - Parameters

somSelf (WPRPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPRPrinter.

pBuf (PSZ) - output
Buffer in which the local print queue name is returned.

pcbBuf (PULONG) - in/out
Pointer to the length, in bytes, of the *pBuf* buffer.

If *pBuf* points to zero, or this call fails, the length of the required buffer plus one is returned in this parameter.

rc (BOOL) - returns
Success indicator.

If the buffer is too small, *pcbBuf* will contain the required size of the buffer. If there does not exist a local alias for this remoter printer object, *pcbBuf* will contain zero.

TRUE
Successful completion.
FALSE
Error occurred.

wpQueryLocalAlias - Remarks

The returned local print queue name can be used to submit print jobs to the network printer. If there is no local print queue that references this network printer object, *pcbBuf* is set to 0 and FALSE is returned.

wpQueryLocalAlias - How to Override

This method is generally not overridden.

wpQueryLocalAlias - Related Methods

Related Methods

- wpQueryNetIdentity
- wpQueryObjectNetId
- wpQuerySrvrIdentity
- wpSetObjectNetId

wpQueryLocalAlias - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpQueryLockupAutoDim

wpQueryLockupAutoDim - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns the current value of the lockup auto-dim option.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL           rc;       /*  Current value of the lockup auto-dim option. */

rc = _wpQueryLockupAutoDim(somSelf);
```

wpQueryLockupAutoDim Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpQueryLockupAutoDim Return Value - rc

rc (BOOL) - returns
Current value of the lockup auto-dim option.

TRUE	Clears the screen and displays the floating lock after the timeout.
FALSE	Does not clear the screen or display the floating lock after the timeout.

wpQueryLockupAutoDim - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

rc (BOOL) - returns
Current value of the lockup auto-dim option.

TRUE	Clears the screen and displays the floating lock after the timeout.
FALSE	Does not clear the screen or display the floating lock after the timeout.

wpQueryLockupAutoDim - Usage

This method can be called at any time in order to determine the lockup automatic dim option for the current desktop.

wpQueryLockupAutoDim - How to Override

This method is generally not overridden.

wpQueryLockupAutoDim - Related Methods

Related Methods

- wpQueryAutoLockup
- wpQueryLockupBackground
- wpQueryLockupFullScreen
- wpQueryLockupOnStartup
- wpQueryLockupTimeout
- wpSetAutoLockup
- wpSetLockupAutoDim
- wpSetLockupBackground
- wpSetLockupFullScreen
- wpSetLockupOnStartup
- wpSetLockupTimeout

wpQueryLockupAutoDim - Example Code

This example lets you find the current Desktop object and retrieves the current lockup automatic dim option setting.

```
WPDesktop    *Desktop;
BOOL         bAutoDim;

Desktop  = _wpclsQueryObjectFromPath("<WP_DESKTOP>");
bAutoDim = _wpQueryLockupAutoDim(Desktop);
```

wpQueryLockupAutoDim - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryLockupBackground

wpQueryLockupBackground - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns the current values for the lockup background.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;            /*  Pointer to the object on which the method is being invoked. */
PSZ           *ppszImageFileName;  /*  Pointer to the name of the background image file (or NULL if none is defi
ULONG         *pulImageMode;       /*  Pointer to one of the following image modes: */
```

```
ULONG          *pulScaleFactor;     /*  Pointer to the scaling factor if the image mode is SCALED_IMAGE. */
ULONG          *pulBackgroundType;  /*  Pointer to one of the following background types: */
LONG           *plBackgroundColor;  /*  Pointer to the background color as an encoded RGB value. */

wpQueryLockupBackground(somSelf, ppszImageFileName,
       pulImageMode, pulScaleFactor, pulBackgroundType,
       plBackgroundColor);
```

wpQueryLockupBackground Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpQueryLockupBackground Parameter - ppszImageFileName

ppszImageFileName (PSZ *) - in/out
Pointer to the name of the background image file (or NULL if none is defined).

wpQueryLockupBackground Parameter - pulImageMode

pulImageMode (ULONG *) - in/out
Pointer to one of the following image modes:

BACKGROUND_NORMAL_IMAGE Normal image.
BACKGROUND_TILED_IMAGE Tiled image.
BACKGROUND_SCALED_IMAGE Scaled image.

wpQueryLockupBackground Parameter - pulScaleFactor

pulScaleFactor (ULONG *) - in/out
Pointer to the scaling factor if the image mode is SCALED_IMAGE.

wpQueryLockupBackground Parameter - pulBackgroundType

pulBackgroundType (ULONG *) - in/out
Pointer to one of the following background types:

BACKGROUND_COLOR_ONLY Color only, no image.

BACKGROUND_IMAGE Type of image (see image modes).

wpQueryLockupBackground Parameter - plBackgroundColor

plBackgroundColor (LONG *) - in/out
Pointer to the background color as an encoded RGB value.

The RGB value is presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

wpQueryLockupBackground - Return Value

The values are returned in the areas pointed to by the individual parameters.

wpQueryLockupBackground - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

ppszImageFileName (PSZ *) - in/out
Pointer to the name of the background image file (or NULL if none is defined).

pulImageMode (ULONG *) - in/out
Pointer to one of the following image modes:

BACKGROUND_NORMAL_IMAGE Normal image.
BACKGROUND_TILED_IMAGE Tiled image.
BACKGROUND_SCALED_IMAGE Scaled image.

pulScaleFactor (ULONG *) - in/out
Pointer to the scaling factor if the image mode is SCALED_IMAGE.

pulBackgroundType (ULONG *) - in/out
Pointer to one of the following background types:

BACKGROUND_COLOR_ONLY Color only, no image.
BACKGROUND_IMAGE Type of image (see image modes).

plBackgroundColor (LONG *) - in/out
Pointer to the background color as an encoded RGB value.

The RGB value is presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

The values are returned in the areas pointed to by the individual parameters.

wpQueryLockupBackground - Usage

This method can be called at any time in order to determine the lockup background for the current desktop.

wpQueryLockupBackground - How to Override

This method is generally not overridden.

wpQueryLockupBackground - Related Methods

Related Methods

- wpQueryAutoLockup
- wpQueryLockupAutoDim
- wpQueryLockupFullScreen
- wpQueryLockupOnStartup
- wpQueryLockupTimeout
- wpSetAutoLockup
- wpSetLockupAutoDim
- wpSetLockupBackground
- wpSetLockupFullScreen
- wpSetLockupOnStartup
- wpSetLockupTimeout

wpQueryLockupBackground - Example Code

This example lets you find the current Desktop object and retrieves the current lockup background definition.

```
WPDesktop    *Desktop;
PSZ          pszImageFile;
ULONG        ulImageMode;
ULONG        ulScaleFactor;
ULONG        ulBackgroundType;
LONG         lBackgroundColor;

Desktop = _wpclsQueryObjectFromPath("<WP_DESKTOP>");
_wpQueryLockupBackground(Desktop,
                         &pszImageFileName,
                         &ulImageMode,
                         &ulScaleFactor,
                         &ulBackgroundType,
                         &lBackgroundColor);
/* pszImageFileName contains the pointer to the name of the image file
 * ulImageMode is BACKGROUND_NORMAL_IMAGE,
 *                BACKGROUND_TILED_IMAGE,
 *                or BACKGROUND_SCALED_IMAGE
 * ulScaleFactor is a number between 1 and 20
 * ulBackgroundType is BACKGROUND_COLOR_ONLY or BACKGROUND_IMAGE
 * lBackgroundColor is an encoded RGB color value
 */
```

wpQueryLockupBackground - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryLockupFullScreen

wpQueryLockupFullScreen - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns the current value of the lockup full-screen option.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL           rc;       /*  Lockup full-screen option. */

rc = _wpQueryLockupFullScreen(somSelf);
```

wpQueryLockupFullScreen Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpQueryLockupFullScreen Return Value - rc

rc (BOOL) - returns
Lockup full-screen option.

TRUE	Displays the full-screen lockup bitmap when locked.
FALSE	Displays the lockup window when locked.

wpQueryLockupFullScreen - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

rc (BOOL) - returns
Lockup full-screen option.

TRUE	Displays the full-screen lockup bitmap when locked.
FALSE	Displays the lockup window when locked.

wpQueryLockupFullScreen - Usage

This method can be called at any time in order to determine the lockup full-screen option for the current desktop.

wpQueryLockupFullScreen - How to Override

This method is generally not overridden.

wpQueryLockupFullScreen - Related Methods

Related Methods

- wpQueryAutoLockup
- wpQueryLockupAutoDim
- wpQueryLockupBackground
- wpQueryLockupOnStartup
- wpQueryLockupTimeout
- wpSetAutoLockup
- wpSetLockupAutoDim
- wpSetLockupBackground
- wpSetLockupFullScreen
- wpSetLockupOnStartup
- wpSetLockupTimeout

wpQueryLockupFullScreen - Example Code

This example lets you find the current Desktop object and retrieves the current lockup, full-screen option setting.

```
WPDesktop    *Desktop;
BOOL         bFullScreen;

Desktop     = _wpclsQueryObjectFromPath("<WP_DESKTOP>");
bFullScreen = _wpQueryLockupFullScreen(Desktop);
```

wpQueryLockupFullScreen - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryLockupOnStartup

wpQueryLockupOnStartup - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns the current value of the lockup on the startup option.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL           rc;       /*  Lockup value. */

rc = _wpQueryLockupOnStartup(somSelf);
```

wpQueryLockupOnStartup Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpQueryLockupOnStartup Return Value - rc

rc (BOOL) - returns
Lockup value.

TRUE	Automatically locks up the system when the system is started.
FALSE	The system is not locked up when started.

wpQueryLockupOnStartup - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

rc (BOOL) - returns
Lockup value.

TRUE	Automatically locks up the system when the system is started.
FALSE	The system is not locked up when started.

wpQueryLockupOnStartup - Usage

This method can be called at any time in order to determine the lockup at startup option for the current desktop.

wpQueryLockupOnStartup - How to Override

This method is generally not overridden.

wpQueryLockupOnStartup - Related Methods

Related Methods

- wpQueryAutoLockup
- wpQueryLockupAutoDim
- wpQueryLockupBackground
- wpQueryLockupFullScreen
- wpQueryLockupTimeout
- wpSetAutoLockup
- wpSetLockupAutoDim
- wpSetLockupBackground
- wpSetLockupFullScreen
- wpSetLockupOnStartup
- wpSetLockupTimeout

wpQueryLockupOnStartup - Example Code

This examples lets you find the current Desktop object and retrieves the current lockup at startup option setting.

```
WPDesktop    *Desktop;
BOOL         bLockupAtStartup;

Desktop = _wpclsQueryObjectFromPath("<WP_DESKTOP>");
bLockupAtStartup = _wpQueryLockupOnStartup(Desktop);
```

wpQueryLockupOnStartup - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryLockupTimeout

wpQueryLockupTimeout - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns the current timeout value for automatic lockup.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG          rc;       /*  Timeout value in minutes. */

rc = _wpQueryLockupTimeout(somSelf);
```

wpQueryLockupTimeout Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpQueryLockupTimeout Return Value - rc

rc (ULONG) - returns
Timeout value in minutes.

wpQueryLockupTimeout - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

rc (ULONG) - returns
Timeout value in minutes.

wpQueryLockupTimeout - Usage

This method can be called at any time in order to determine the lockup automatic timeout value for the current desktop.

wpQueryLockupTimeout - How to Override

This method is generally not overridden.

wpQueryLockupTimeout - Related Methods

Related Methods

- wpQueryAutoLockup
- wpQueryLockupAutoDim
- wpQueryLockupBackground
- wpQueryLockupFullScreen
- wpQueryLockupOnStartup
- wpSetAutoLockup
- wpSetLockupAutoDim
- wpSetLockupBackground
- wpSetLockupFullScreen
- wpSetLockupOnStartup
- wpSetLockupTimeout

wpQueryLockupTimeout - Example Code

This example lets you find the current Desktop object and retrieves the current lockup automatic timeout value setting.

```
WPDesktop    *Desktop;
ULONG        ulTimeout;

Desktop = _wpclsQueryObjectFromPath("<WP_DESKTOP>");
ulTimeout = _wpQueryLockupTimeout(Desktop);
```

wpQueryLockupTimeout - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryLogicalDrive

wpQueryLogicalDrive - Syntax

This instance method returns the logical drive number that is represented by this disk object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDisk     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
ULONG       ulDiskID;  /*  Logical drive identifier. */

ulDiskID = _wpQueryLogicalDrive(somSelf);
```

wpQueryLogicalDrive Parameter - somSelf

somSelf (WPDisk *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDisk.

wpQueryLogicalDrive Return Value - ulDiskID

ulDiskID (ULONG) - returns
Logical drive identifier.

wpQueryLogicalDrive - Parameters

somSelf (WPDisk *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDisk.

ulDiskID (ULONG) - returns
Logical drive identifier.

wpQueryLogicalDrive - Remarks

Every instance of the WPDisk class that is created in the system must represent a logical drive partition. There should never be more than one disk object per logical drive.

wpQueryLogicalDrive - Usage

This method can be called at any time.

wpQueryLogicalDrive - How to Override

This method should not be overridden.

wpQueryLogicalDrive - Example Code

```
 ULONG           ulLogicalDriveNum = 0;   /* Logical drive number returned */
 WPRootFolder   *RootFolder;              /* Root folder for drive         */

  /*
   *  Query logical drive number for the WPDisk object
   */

   ulLogicalDriveNum = _wpQueryLogicalDrive( self );


   if (LogicalDrive == 0) {

     somPrintf("_wpQueryLogicalDrive failed");
     return 1;

   } /* endif */


figseg fit=12.
  /*
   *  Query root folder also
   */

   RootFolder = _wpQueryRootFolder( self );

   if (RootFolder == NULL) {

     somPrintf("_wpQueryRootFolder failed");
     return 1;

   } /* endif */
```

wpQueryLogicalDrive - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Glossary

wpQueryMenuBarVisibility

wpQueryMenuBarVisibility - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns the visibility state of the folder's menu bar.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         rc;       /*  Visibility state. */

rc = _wpQueryMenuBarVisibility(somSelf);
```

wpQueryMenuBarVisibility Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryMenuBarVisibility Return Value - rc

rc (ULONG) - returns
Visibility state.

MENUBAR_ON	Menu bar is visible.
MENUBAR_OFF	Menu bar is invisible.

wpQueryMenuBarVisibility - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

rc (ULONG) - returns
Visibility state.

MENUBAR_ON	Menu bar is visible.
MENUBAR_OFF	Menu bar is invisible.

wpQueryMenuBarVisibility - Usage

This method can be called at any time in order to determine whether or not an open view of this folder displays a menu bar.

wpQueryMenuBarVisibility - How to Override

This method is generally not overridden.

wpQueryMenuBarVisibility - Related Methods

Related Methods

- wpSetMenuBarVisibility

wpQueryMenuBarVisibility - Example Code

This example determines whether or not this folder displays the menu bars.

```
ULONG        ulMenuBarOption;

ulMenuBarOption = _wpQueryMenuBarVisibility(somSelf);
/* ulMenuBarOption is MENUBAR_ON or MENUBAR_OFF */
```

wpQueryMenuBarVisibility - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryMenuStyle

wpQueryMenuStyle - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns the current menu style, which is either long or short.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         ulStyle;  /*  Menu style. */

ulStyle = _wpQueryMenuStyle(somSelf);
```

wpQueryMenuStyle Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryMenuStyle Return Value - ulStyle

ulStyle (ULONG) - returns
Menu style.

MENUS_LONG	Long menus.
MENUS_SHORT	Short menus.

wpQueryMenuStyle - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulStyle (ULONG) - returns
Menu style.

MENUS_LONG	Long menus.
MENUS_SHORT	Short menus.

wpQueryMenuStyle - Usage

This method can be called at any time in order to determine whether this object displays long or short pop-up menus.

wpQueryMenuStyle - How to Override

This method is generally not overridden.

wpQueryMenuStyle - Related Methods

Related Methods

- wpSetMenuStyle

wpQueryMenuStyle - Example Code

This example determines whether this object displays long or short pop-up menus.

```
ULONG        ulMenuStyle;

ulMenuStyle = _wpQueryMenuStyle(somSelf);
/* ulMenuStyle is MENUS_LONG or MENUS_SHORT */
```

wpQueryMenuStyle - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods

wpQueryMinWindow

wpQueryMinWindow - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the minimized window behavior of an object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
ULONG         ulMinWin;  /*  Flag indicating minimized window behavior. */

ulMinWin = _wpQueryMinWindow(somSelf);
```

wpQueryMinWindow Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryMinWindow Return Value - ulMinWin

ulMinWin (ULONG) - returns
Flag indicating minimized window behavior.

Possible values are described in the following list:

MINWIN_DEFAULT
System default.
MINWIN_HIDDEN
Minimized windows are hidden.
MINWIN_VIEWER
Minimized windows are placed in the viewer.
MINWIN_DESKTOP
Minimized windows are placed on the desktop.

wpQueryMinWindow - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulMinWin (ULONG) - returns
Flag indicating minimized window behavior.

Possible values are described in the following list:

MINWIN_DEFAULT
System default.
MINWIN_HIDDEN
Minimized windows are hidden.
MINWIN_VIEWER
Minimized windows are placed in the viewer.
MINWIN_DESKTOP
Minimized windows are placed on the desktop.

wpQueryMinWindow - Usage

This method is usually called by the system in order to retrieve an object's minimized window behavior.

wpQueryMinWindow - How to Override

This method is generally not overridden.

wpQueryMinWindow - Related Methods

Related Methods

- wpSetMinWindow

wpQueryMinWindow - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpQueryNameClashOptions

wpQueryNameClashOptions - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the options that will not be available on the name clash dialog.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
ULONG         menuID;       /*  Menu ID of selected menu item that caused name clash options to be queried. */
ULONG         ulNameClash;  /*  Flag indicating options that are to be unavailable. */

ulNameClash = _wpQueryNameClashOptions(somSelf,
                menuID);
```

wpQueryNameClashOptions Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryNameClashOptions Parameter - menuID

menuID (ULONG) - input
Menu ID of selected menu item that caused name clash options to be queried.

For a listing of WPMENUIDs, see the individual object classes.

wpQueryNameClashOptions Return Value - ulNameClash

ulNameClash (ULONG) - returns
Flag indicating options that are to be unavailable.

Possible values are described in the following list:

NO_NAMECLASH_RENAME
Renames are not allowed.
NO_NAMECLASH_DIALOG
Dialogs are not shown.
NO_NAMECLASH_APPEND
Appends are not allowed.
NO_NAMECLASH_REPLACE

Replacements are not allowed.

wpQueryNameClashOptions - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

menuID (ULONG) - input
Menu ID of selected menu item that caused name clash options to be queried.

For a listing of WPMENUIDs, see the individual object classes.

ulNameClash (ULONG) - returns
Flag indicating options that are to be unavailable.

Possible values are described in the following list:

NO_NAMECLASH_RENAME
Renames are not allowed.
NO_NAMECLASH_DIALOG
Dialogs are not shown.
NO_NAMECLASH_APPEND
Appends are not allowed.
NO_NAMECLASH_REPLACE
Replacements are not allowed.

wpQueryNameClashOptions - How to Override

This method is generally not overridden.

wpQueryNameClashOptions - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Glossary

wpQueryNetIdentity

wpQueryNetIdentity - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the fully qualified network name of this network group.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPNetgrp     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
PSZ           pszNetIdentity;  /*  Fully qualified network name of this network group. */

pszNetIdentity = _wpQueryNetIdentity(somSelf);
```

wpQueryNetIdentity Parameter - somSelf

somSelf (WPNetgrp *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPNetgrp.

wpQueryNetIdentity Return Value - pszNetIdentity

pszNetIdentity (PSZ) - returns
Fully qualified network name of this network group.

For example, LSNetwork.

wpQueryNetIdentity - Parameters

somSelf (WPNetgrp *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPNetgrp.

pszNetIdentity (PSZ) - returns
Fully qualified network name of this network group.

For example, LSNetwork.

wpQueryNetIdentity - How to Override

This method is generally not overridden.

wpQueryNetIdentity - Related Methods

Related Methods

- wpQueryObjectNetId
- wpQuerySrvrIdentity
- wpSetNetIdentity
- wpSetObjectNetId

wpQueryNetIdentity - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryNetworkId

wpQueryNetworkId - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the fully qualified network ID of the remote print queue for the network printer object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPRPrinter     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PSZ             pBuf;     /*  Buffer in which the network ID is returned. */
PULONG          pcbBuf;   /*  Size, in bytes, of the pBuf buffer. */
BOOL            rc;       /*  Success indicator. */

rc = _wpQueryNetworkId(somSelf, pBuf, pcbBuf);
```

wpQueryNetworkId Parameter - somSelf

somSelf (WPRPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPRPrinter.

wpQueryNetworkId Parameter - pBuf

pBuf (PSZ) - output
Buffer in which the network ID is returned.

The network ID must be in the following form:

```
<network>:\\<server>\<resource>
```

For example, LS:\\DEPTSERV\DEPTPRNT.

A return value of NULL indicates that *pcbBuf* is too small to hold the network ID.

wpQueryNetworkId Parameter - pcbBuf

pcbBuf (PULONG) - in/out
Size, in bytes, of the *pBuf* buffer.

If *pBuf* is NULL or this call fails, the length of the required buffer is returned in this parameter.

wpQueryNetworkId Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
: Successful completion.

FALSE
: Error occurred.

wpQueryNetworkId - Parameters

somSelf (WPRPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPRPrinter.

pBuf (PSZ) - output
Buffer in which the network ID is returned.

The network ID must be in the following form:

```
<network>:\\<server>\<resource>
```

For example, LS:\\DEPTSERV\DEPTPRNT.

A return value of NULL indicates that *pcbBuf* is too small to hold the network ID.

pcbBuf (PULONG) - in/out
Size, in bytes, of the *pBuf* buffer.

If *pBuf* is NULL or this call fails, the length of the required buffer is returned in this parameter.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.

FALSE
Error occurred.

wpQueryNetworkId - How to Override

This method is generally not overridden.

wpQueryNetworkId - Related Methods

Related Methods

- wpQueryNetIdentity
- wpQueryObjectNetId
- wpQuerySrvrIdentity
- wpSetObjectNetId

wpQueryNetworkId - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryNextIconPos

wpQueryNextIconPos - Syntax

This instance method is called to allow the folder to query the next icon position.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
```

```
PPOINTL       pptl;     /*  Pointer to the next position at which icons will be inserted. */

pptl = _wpQueryNextIconPos(somSelf);
```

wpQueryNextIconPos Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryNextIconPos Return Value - pptl

pptl (PPOINTL) - returns
Pointer to the next position at which icons will be inserted.

wpQueryNextIconPos - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

pptl (PPOINTL) - returns
Pointer to the next position at which icons will be inserted.

wpQueryNextIconPos - Remarks

The next icon is the next available *parking space* within the folder's client area.

wpQueryNextIconPos - Usage

This method can be called at any time in order to determine the next position at which objects will be inserted in the file.

wpQueryNextIconPos - How to Override

This method is generally not overridden.

wpQueryNextIconPos - Related Methods

Related Methods

- wpSetNextIconPos

wpQueryNextIconPos - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpQueryObjectID

wpQueryObjectID - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the unique, text object ID that was assigned to the given object when it was installed or changed (with wpSetup).

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
PSZ           pszObjectID;  /*  String containing a unique object ID. */

pszObjectID = _wpQueryObjectID(somSelf);
```

wpQueryObjectID Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryObjectID Return Value - pszObjectID

pszObjectID (PSZ) - returns
String containing a unique object ID.

For example, "<WP_DESKTOP>".

wpQueryObjectID - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pszObjectID (PSZ) - returns
String containing a unique object ID.

For example, "<WP_DESKTOP>".

wpQueryObjectID - Remarks

The install routine uses this variable to ensure that a given object is never re-installed, thereby avoiding duplication of objects.

The object ID is always lost when the object is copied; therefore, only one object ever exists for a given object ID.

wpQueryObjectID - How to Override

This method is generally not overridden.

wpQueryObjectID - Related Methods

Related Methods

- wpSetObjectID

wpQueryObjectID - Topics

Select an item:
Syntax
Parameters
Returns

Remarks
How to Override
Related Methods
Glossary

wpQueryObjectList

wpQueryObjectList - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns a list of object buttons displayed on the Toolbar or drawer.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;        /*  Pointer to the object on which the method is being invoked. */
ULONG            ulDrawer;       /*  Toolbar or drawer being queried. */
PULONG           pulNumObjects;  /*  Pointer to the number of objects returned. */
HOBJECT         *hObject;        /*  Pointer to an array of objects displayed on the Toolbar or drawers. */

hObject = _wpQueryObjectList(somSelf, ulDrawer,
            pulNumObjects);
```

wpQueryObjectList Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpQueryObjectList Parameter - ulDrawer

ulDrawer (ULONG) - input
Toolbar or drawer being queried.

0
Toolbar.
>0
Drawer identifier.

wpQueryObjectList Parameter - pulNumObjects

pulNumObjects (PULONG) - output
Pointer to the number of objects returned.

wpQueryObjectList Return Value - hObject

hObject (HOBJECT *) - returns
Pointer to an array of objects displayed on the Toolbar or drawers.

wpQueryObjectList - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

ulDrawer (ULONG) - input
Toolbar or drawer being queried.

0
Toolbar.
>0
Drawer identifier.

pulNumObjects (PULONG) - output
Pointer to the number of objects returned.

hObject (HOBJECT *) - returns
Pointer to an array of objects displayed on the Toolbar or drawers.

wpQueryObjectList - How to Override

This method is generally not overridden.

wpQueryObjectList - Related Methods

Related Methods

- wpRefreshDrawer

wpQueryObjectList - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryObjectNetId

wpQueryObjectNetId - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the network ID of the Shared Directory object that this object is linked to.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPNetLink     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
PSZ            pszNetIdentity;  /*  Pointer to the network ID string. */

pszNetIdentity = _wpQueryObjectNetId(somSelf);
```

wpQueryObjectNetId Parameter - somSelf

somSelf (WPNetLink *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPNetLink.

wpQueryObjectNetId Return Value - pszNetIdentity

pszNetIdentity (PSZ) - returns
Pointer to the network ID string.

The network name is in the following format:

```
"<Network>\<Server>\<Share Name>"
```

Where:

<Network>	Network identification
<Server>	Server identification
<Share Name>	

wpQueryObjectNetId - Parameters

somSelf (WPNetLink *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPNetLink.

pszNetIdentity (PSZ) - returns
Pointer to the network ID string.

The network name is in the following format:

```
"<Network>\<Server>\<Share Name>"
```

Where:

<Network>	Network identification
<Server>	Server identification
<Share Name>	

wpQueryObjectNetId - Remarks

The Shared Directory object uses this ID to make its network calls. This ID is not stored in the object's data.

wpQueryObjectNetId - Usage

This method can be called at any time to retrieve the network ID of an object.

wpQueryObjectNetId - How to Override

This method is generally not overridden.

wpQueryObjectNetId - Related Methods

Related Methods

- wpQueryNetIdentity
- wpQuerySrvrIdentity
- wpSetObjectNetId

wpQueryObjectNetId - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpQueryOpenFolders

wpQueryOpenFolders - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method allows for the enumeration of all open folders in the system.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
ULONG         ulOption;  /*  Flag indicating the type of operation. */
WPObject     *Object;    /*  Pointer to an open folder object. */

Object = _wpQueryOpenFolders(somSelf, ulOption);
```

wpQueryOpenFolders Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryOpenFolders Parameter - ulOption

ulOption (ULONG) - input
Flag indicating the type of operation.

QC_FIRST	Returns the first item in the open folder list.
QC_NEXT	Returns the next item in the open folder list.

wpQueryOpenFolders Return Value - Object

Object (WPObject *) - returns
Pointer to an open folder object.

wpQueryOpenFolders - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulOption (ULONG) - input
Flag indicating the type of operation.

QC_FIRST	Returns the first item in the open folder list.
QC_NEXT	Returns the next item in the open folder list.

Object (WPObject *) - returns
Pointer to an open folder object.

wpQueryOpenFolders - Remarks

OS/2 maintains a list of all open folders in the system. wpQueryOpenFolders provides access to this list.

The first time you call wpQueryOpenFolders specify QC_FIRST in *ulOption*. A pointer to the first open folder in the open folder list is returned in *Object*. On subsequent calls specify QC_NEXT in *ulOption* and the pointer to the previously returned folder in *Object*. A pointer to the next folder in the open folder list is returned. wpQueryOpenFolders returns NULL after the last folder in the list has been returned.

wpQueryOpenFolders returns the next open folder after the folder passed in *Object*, or a NULL if the folder passed in *Object* is the last folder in the open folder list.

If wpQueryOpenFolders is called with QC_NEXT in *ulOption* and a pointer to a closed folder in *Object* then a QC_FIRST operation is performed.

Note: Do not close the open views of a folder returned by the wpQueryOpenFolders method because this breaks the chain of open folders. If you close the last open view of a folder, the next call to wpQueryOpenFolders will return NULL, even though there are other open folders in the system.

wpQueryOpenFolders - Usage

You can call the wpQueryOpenFolders method in a loop to enumerate all open folders in the system.

wpQueryOpenFolders - How to Override

This method is generally not be overridden.

wpQueryOpenFolders - Related Methods

Related Methods

- wpclsQueryFolder

wpQueryOpenFolders - Example Code

In this example, we enumerate all open folders in the system. In the first call to the wpQueryOpenFolders method, we must pass the pointer to any folder. In the subsequent calls to the wpQueryOpenFolders method, we must pass the pointer to the folder returned by the previous call to the wpQueryOpenFolders method.

```
WPFolder    *Folder;

/* Find the first open folder  */
Folder = _wpQueryOpenFolders(somSelf,QC_FIRST);
while (Folder)
{

   /***** Code to process the folder goes here *****/

   /* Find the next open folder   */
   Folder = _wpQueryOpenFolders(Folder,QC_NEXT);
}
```

wpQueryOpenFolders - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryPaletteHelp

wpQueryPaletteHelp - Syntax

This instance method returns the help panel ID that is displayed when the help pushbutton is used from an open palette view.

```
#define INCL_WINWORKPLACE
```

```
#include <os2.h>

WPPalette     *somSelf;        /*  Pointer to the object on which the method is being invoked. */
ULONG          ulHelpPanelID;  /*  Help panel ID. */

ulHelpPanelID = _wpQueryPaletteHelp(somSelf);
```

wpQueryPaletteHelp Parameter - somSelf

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

wpQueryPaletteHelp Return Value - ulHelpPanelID

ulHelpPanelID (ULONG) - returns
Help panel ID.

The help panel ID within help module of this class is specified by the wpQueryDefaultHelp method for this object instance.

wpQueryPaletteHelp - Parameters

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

ulHelpPanelID (ULONG) - returns
Help panel ID.

The help panel ID within help module of this class is specified by the wpQueryDefaultHelp method for this object instance.

wpQueryPaletteHelp - Remarks

This method returns the help panel ID that is visible from the open palette view. That panel should describe what the palette cells represent, how to edit them, and how to apply the cell values to other windows or objects. The palette object can specify class-default and instance-specific helps for the object in addition to this specialized help which applies only to the palette view.

wpQueryPaletteHelp - Usage

This method may be called at any time.

wpQueryPaletteHelp - How to Override

All subclasses of WPPalette need to override this method to provide help about their open palette view window.

wpQueryPaletteHelp - Related Methods

Related Methods

- wpQueryDefaultHelp
- wpclsQueryDefaultHelp
- wpSetDefaultHelp

wpQueryPaletteHelp - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpQueryPaletteInfo

wpQueryPaletteInfo - Syntax

This instance method is called to determine current information about the palette.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPalette     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
PPALINFO       pPalInfo;  /*  Pointer to a PALINFO structure. */
BOOL           rc;        /*  Success indicator. */

rc = _wpQueryPaletteInfo(somSelf, pPalInfo);
```

wpQueryPaletteInfo Parameter - somSelf

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

wpQueryPaletteInfo Parameter - pPalInfo

pPalInfo (PPALINFO) - input
Pointer to a PALINFO structure.

wpQueryPaletteInfo Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryPaletteInfo - Parameters

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

pPalInfo (PPALINFO) - input
Pointer to a PALINFO structure.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryPaletteInfo - Remarks

The palette information can be set by issuing a call to the wpSetPaletteInfo method.

wpQueryPaletteInfo - Usage

This method can be called at any time in order to get current information about the palette.

wpQueryPaletteInfo - How to Override

This method is generally not overridden.

wpQueryPaletteInfo - Related Methods

Related Methods

- wpSetPaletteInfo

wpQueryPaletteInfo - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpQueryPointerFromContents

wpQueryPointerFromContents - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns a pointer to shared memory containing the contents of the WPDataFile object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDataFile     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PVOID           pData;    /*  Pointer to shared memory if successful, or NULL if unsuccessful. */

pData = _wpQueryPointerFromContents(somSelf);
```

wpQueryPointerFromContents Parameter - somSelf

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

wpQueryPointerFromContents Return Value - pData

pData (PVOID) - returns
Pointer to shared memory if successful, or NULL if unsuccessful.

The first word is a ULONG that contains the length, followed immediately by the data.

wpQueryPointerFromContents - Parameters

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

pData (PVOID) - returns
Pointer to shared memory if successful, or NULL if unsuccessful.

The first word is a ULONG that contains the length, followed immediately by the data.

wpQueryPointerFromContents - Remarks

The caller of the wpQueryPointerFromContents method is responsible to free the returned memory segment by calling the DosFreeMem function.

wpQueryPointerFromContents - Usage

This method can be called at any time to objtain a pointer to a memory segment containing the contents of the data file.

wpQueryPointerFromContents - How to Override

This method is not generally overridden.

wpQueryPointerFromContents - Related Methods

Related Methods

- wpQueryHandleFromContents
- wpSetContentsFromHandle
- wpSetContentsFromPointer

wpQueryPointerFromContents - Example Code

This example shows how to obtain the data from a data file.

```
PVOID        pData;

/* Get the data from the data file  */
pData = _wpQueryPointerFromContents(somSelf);

/****** Process the data from the data file *****/

/* Free the shared memory object     */
DosFreeMem(pData);
```

wpQueryPointerFromContents - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryPowerConfirmation

wpQueryPowerConfirmation - Syntax

This method is specific to version 2.1, or higher, of the OS/2 operating system.

This instance method queries the current confirmation message control value for the power object. This message is displayed when power state changes are requested from the popup menu of the power object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPower     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL         rc;       /*  Power confirmation flag. */
```

```
rc = _wpQueryPowerConfirmation(somSelf);
```

wpQueryPowerConfirmation Parameter - somSelf

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

wpQueryPowerConfirmation Return Value - rc

rc (BOOL) - returns
Power confirmation flag.

TRUE	The power confirmation is enabled.
FALSE	The power confirmation is disabled.

wpQueryPowerConfirmation - Parameters

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

rc (BOOL) - returns
Power confirmation flag.

TRUE	The power confirmation is enabled.
FALSE	The power confirmation is disabled.

wpQueryPowerConfirmation - Related Methods

Related Methods

- wpAddPowerPage
- wpAddPowerViewPage
- wpChangePowerState
- wpQueryAutoRefresh
- wpQueryPowerConfirmation
- wpQueryPowerManagement
- wpQueryRefreshRate
- wpSetAutoRefresh
- wpSetPowerConfirmation
- wpSetPowerManagement
- wpSetRefreshRate

wpQueryPowerConfirmation - Topics

Select an item:
Syntax
Parameters
Returns
Related Methods
Glossary

wpQueryPowerManagement

wpQueryPowerManagement - Syntax

This method is specific to version 2.1, or higher, of the OS/2 operating system.

This instance method provides the ability to query the enable or disable state of the OS/2 Power Management support.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPower     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL         rc;       /*  Power management flag. */

rc = _wpQueryPowerManagement(somSelf);
```

wpQueryPowerManagement Parameter - somSelf

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

wpQueryPowerManagement Return Value - rc

rc (BOOL) - returns
Power management flag.

TRUE	The power management is enabled.
FALSE	The power management is disabled.

wpQueryPowerManagement - Parameters

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

rc (BOOL) - returns
Power management flag.

TRUE	The power management is enabled.
FALSE	The power management is disabled.

wpQueryPowerManagement - Related Methods

Related Methods

- wpAddPowerPage
- wpAddPowerViewPage
- wpChangePowerState
- wpQueryAutoRefresh
- wpQueryPowerConfirmation
- wpQueryPowerManagement
- wpQueryRefreshRate
- wpSetAutoRefresh
- wpSetPowerConfirmation
- wpSetPowerManagement
- wpSetRefreshRate

wpQueryPowerManagement - Topics

Select an item:
Syntax
Parameters
Returns
Related Methods
Glossary

wpQueryPrinterName

wpQueryPrinterName - Syntax

This instance method is called to query the name of the printer.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPrinter     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
PSZ            pszPrinterName;  /*  Returned printer queue name of the object queried. */
BOOL           rc;              /*  Success indicator. */

rc = _wpQueryPrinterName(somSelf, pszPrinterName);
```

wpQueryPrinterName Parameter - somSelf

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

wpQueryPrinterName Parameter - pszPrinterName

pszPrinterName (PSZ) - output
Returned printer queue name of the object queried.

wpQueryPrinterName Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryPrinterName - Parameters

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

pszPrinterName (PSZ) - output
Returned printer queue name of the object queried.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryPrinterName - Remarks

If this is a printer object, the printer queue name is defined on the computer returned by wpQueryComputerName. Use wpQueryLocalAlias to get the local printer queue name for network printer objects.

wpQueryPrinterName - How to Override

This method is generally not overridden.

wpQueryPrinterName - Related Methods

Related Methods

- wpQueryComputerName
- wpSetPrinterName
- wpQueryLocalAlias

wpQueryPrinterName - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpQueryPrintObject

wpQueryPrintObject - Syntax

This method is specific to Version 3, or higher, of the OS/2 operating system.

This instance method returns a pointer to the printer object that the current job is in.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPJob        *somSelf;  /*  Pointer to the object on which the method is being invoked. */
WPObject     *Object;   /*  Pointer to the printer object. */
```

```
Object = _wpQueryPrintObject(somSelf);
```

wpQueryPrintObject Parameter - somSelf

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

wpQueryPrintObject Return Value - Object

Object (WPObject *) - returns
Pointer to the printer object.

wpQueryPrintObject - Parameters

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

Object (WPObject *) - returns
Pointer to the printer object.

wpQueryPrintObject - Usage

This method can be called at any time.

wpQueryPrintObject - How to Override

This method is generally not overridden.

wpQueryPrintObject - Topics

Select an item:
Syntax

Parameters
Returns
How to Override
Usage
Glossary

wpQueryProgDetails

wpQueryProgDetails - Syntax

This instance method is called to allow the object to query its program details.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgram      *somSelf;       /*  Pointer to the object on which the method is being invoked. */
PPROGDETAILS    pProgDetails;  /*  Pointer to the program details. */
PULONG          pulSize;       /*  Size of the pProgDetails buffer. */
BOOL            rc;            /*  Success indicator. */

rc = _wpQueryProgDetails(somSelf, pProgDetails,
       pulSize);
```

wpQueryProgDetails Parameter - somSelf

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

wpQueryProgDetails Parameter - pProgDetails

pProgDetails (PPROGDETAILS) - input
Pointer to the program details.

wpQueryProgDetails Parameter - pulSize

pulSize (PULONG) - in/out
Size of the *pProgDetails* buffer.

If NULL is specified for *pProgDetails* , then the size of the current *pProgDetails* is returned in *pulSize* .

wpQueryProgDetails Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryProgDetails - Parameters

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

pProgDetails (PPROGDETAILS) - input
Pointer to the program details.

pulSize (PULONG) - in/out
Size of the *pProgDetails* buffer.

If NULL is specified for *pProgDetails* , then the size of the current *pProgDetails* is returned in *pulSize* .

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryProgDetails - Usage

This method can be called at any time in order to determine the details of this object.

wpQueryProgDetails - How to Override

This method is generally not overridden.

wpQueryProgDetails - Related Methods

Related Methods

- wpQueryProgDetails (WPProgramFile)
- wpSetProgDetails (WPProgram)

- wpSetProgDetails (WPProgramFile)

wpQueryProgDetails - Example Code

```
PPROGDETAILS pProgDetails;
ULONG        ulSize;


          /* Get information about a program object */

  if ((_wpQueryProgDetails (self, (PPROGDETAILS)NULL, &ulSize)))
    {
    if ((pProgDetails = (PPROGDETAILS) _wpAllocMem( self, ulSize, NULL)) != NULL)
      {
      if ((_wpQueryProgDetails (self, pProgDetails, &ulSize)))
        {
         somPrintf("_wpQueryProgDetails reports:\n");
         somLPrintf(1,"pszExecutable = %s\n", pProgDetails->pszExecutable);
         somLPrintf(1,"pszParameters = %s\n", pProgDetails->pszParameters);
        }

        else
          {
          somPrintf("_wpQueryProgDetails - unable to get details.\n");
          return 1;
          }
        }
      else
        {
        somPrintf("_wpAllocMem error\n");
        return 1;
        }
      }
    else
      {
      somPrintf("_wpQueryProgDetails - unable to determine size for buffer.\n");
      return 1;
      }
```

wpQueryProgDetails - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryProgDetails

wpQueryProgDetails - Syntax

This instance method is called to allow the object to query its program details.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgramFile     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
PPROGDETAILS       pProgDetails;  /*  Pointer to the program details. */
PULONG             pSize;         /*  Size of the pProgDetails buffer. */
BOOL               rc;            /*  Success indicator. */

rc = _wpQueryProgDetails(somSelf, pProgDetails,
       pSize);
```

wpQueryProgDetails Parameter - somSelf

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

wpQueryProgDetails Parameter - pProgDetails

pProgDetails (PPROGDETAILS) - input
Pointer to the program details.

wpQueryProgDetails Parameter - pSize

pSize (PULONG) - in/out
Size of the *pProgDetails* buffer.

If NULL is specified for *pProgDetails* , then the size of the current *pProgDetails* is returned in *pSize* .

wpQueryProgDetails Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryProgDetails - Parameters

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

pProgDetails (PPROGDETAILS) - input
Pointer to the program details.

pSize (PULONG) - in/out
Size of the *pProgDetails* buffer.

If NULL is specified for *pProgDetails* , then the size of the current *pProgDetails* is returned in *pSize* .

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryProgDetails - Usage

This method can be called at any time in order to determine the details of this object.

wpQueryProgDetails - How to Override

This method is generally not overridden.

wpQueryProgDetails - Related Methods

Related Methods

- wpQueryProgDetails (WPProgram)
- wpSetProgDetails (WPProgram)
- wpSetProgDetails (WPProgramFile)
- wpSetProgDetails

wpQueryProgDetails - Example Code

```
PPROGDETAILS pProgDetails;
ULONG        ulSize;
```

```
          /* Get information about a program object */

   if ((_wpQueryProgDetails (self, (PPROGDETAILS)NULL, &ulSize)))
     {
     if ((pProgDetails = (PPROGDETAILS) _wpAllocMem( self, ulSize, NULL)) != NULL)
       {
       if ((_wpQueryProgDetails (self, pProgDetails, &ulSize)))
         {
          somPrintf("_wpQueryProgDetails reports:\n");
          somLPrintf(1,"pszExecutable = %s\n", pProgDetails->pszExecutable);
          somLPrintf(1,"pszParameters = %s\n", pProgDetails->pszParameters);
         }

         else
           {
           somPrintf("_wpQueryProgDetails - unable to get details.\n");
           return 1;
           }
         }
       else
         {
         somPrintf("_wpAllocMem error\n");
         return 1;
         }
       }
     else
       {
       somPrintf("_wpQueryProgDetails - unable to determine size for buffer.\n");
       return 1;
       }
```

wpQueryProgDetails - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryProgramAssociations

wpQueryProgramAssociations - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the program associations from the .INI file.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgram     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
PBYTE          ptr;
PSZ            pszAssoc;      /*  Pointer to string of datafile association types or filters, separated by comma
```

```
BOOL            fFilter;        /*  Filter flag. */
PSZ             ProgramAssoc;

ProgramAssoc = _wpQueryProgramAssociations(
                somSelf, ptr, pszAssoc, fFilter);
```

wpQueryProgramAssociations Parameter - somSelf

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

wpQueryProgramAssociations Parameter - ptr

ptr (PBYTE) - input

wpQueryProgramAssociations Parameter - pszAssoc

pszAssoc (PSZ) - input
Pointer to string of datafile association types or filters, separated by commas.

wpQueryProgramAssociations Parameter - fFilter

fFilter (BOOL) - input
Filter flag.

If TRUE, *pszAssoc* should contain filters, as opposed to types.

TRUE
FALSE

wpQueryProgramAssociations Return Value - ProgramAssoc

ProgramAssoc (PSZ) - returns

wpQueryProgramAssociations - Parameters

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

ptr (PBYTE) - input

pszAssoc (PSZ) - input
Pointer to string of datafile association types or filters, separated by commas.

fFilter (BOOL) - input
Filter flag.

If TRUE, *pszAssoc* should contain filters, as opposed to types.

TRUE
FALSE

ProgramAssoc (PSZ) - returns

wpQueryProgramAssociations - Usage

This method can be called at any time.

wpQueryProgramAssociations - How to Override

This method is generally not overridden.

wpQueryProgramAssociations - Related Methods

Related Methods

- wpSetProgramAssociations

wpQueryProgramAssociations - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods

wpQueryProgramAssociations

wpQueryProgramAssociations - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the program associations from the .INI file.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgramFile     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
PBYTE              ptr;
PSZ                pszAssoc;
BOOL               fFilter;       /*  Filter flag. */
PSZ                ProgramAssoc;

ProgramAssoc = _wpQueryProgramAssociations(
                somSelf, ptr, pszAssoc, fFilter);
```

wpQueryProgramAssociations Parameter - somSelf

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

wpQueryProgramAssociations Parameter - ptr

ptr (PBYTE) - input

wpQueryProgramAssociations Parameter - pszAssoc

pszAssoc (PSZ) - input

wpQueryProgramAssociations Parameter - fFilter

fFilter (BOOL) - input
Filter flag.

TRUE
FALSE

wpQueryProgramAssociations Return Value - ProgramAssoc

ProgramAssoc (PSZ) - returns

wpQueryProgramAssociations - Parameters

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

ptr (PBYTE) - input

pszAssoc (PSZ) - input

fFilter (BOOL) - input
Filter flag.

TRUE
FALSE

ProgramAssoc (PSZ) - returns

wpQueryProgramAssociations - Usage

This method can be called at any time.

wpQueryProgramAssociations - How to Override

This method is generally not overridden.

wpQueryProgramAssociations - Related Methods

Related Methods

- wpSetProgramAssociations

wpQueryProgramAssociations - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpQueryQueueOptions

wpQueryQueueOptions - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the printer object's queue options.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPrinter     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
ULONG          ulOptions;  /*  Flag indicating queue option settings. */

ulOptions = _wpQueryQueueOptions(somSelf);
```

wpQueryQueueOptions Parameter - somSelf

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

wpQueryQueueOptions Return Value - ulOptions

ulOptions (ULONG) - returns
Flag indicating queue option settings.

The following flags can be ORed together:

PO_PRINTERSPECIFIC
If set, the printer object spools print jobs in PM_Q_RAW format. If cleared, the printer object spools print jobs in PM_Q_STD format.

PO_PRINTWHILESPOOLING
If set, printing is enabled while the job is spooling. If cleared, printing occurs only after the job is spooled.

PO_APPDEFAULT
If set, this printer object becomes the application's default printer object. If cleared, the application's default printer object is not changed.

PO_DIALOGBEFOREPRINT
If set, a Job Properties dialog is displayed when a job is submitted to this printer object. If cleared, the Job Properties dialog is not displayed.

wpQueryQueueOptions - Parameters

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

ulOptions (ULONG) - returns
Flag indicating queue option settings.

The following flags can be ORed together:

PO_PRINTERSPECIFIC
If set, the printer object spools print jobs in PM_Q_RAW format. If cleared, the printer object spools print jobs in PM_Q_STD format.

PO_PRINTWHILESPOOLING
If set, printing is enabled while the job is spooling. If cleared, printing occurs only after the job is spooled.

PO_APPDEFAULT
If set, this printer object becomes the application's default printer object. If cleared, the application's default printer object is not changed.

PO_DIALOGBEFOREPRINT
If set, a Job Properties dialog is displayed when a job is submitted to this printer object. If cleared, the Job Properties dialog is not displayed.

wpQueryQueueOptions - Usage

This method can be called at any time.

wpQueryQueueOptions - How to Override

This method is generally not overridden.

wpQueryQueueOptions - Related Methods

Related Methods

- wpSetQueueOptions

wpQueryQueueOptions - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpQueryRealName

wpQueryRealName - Syntax

This instance method is called to allow the object to query its physical file name.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
PSZ               pszFilename;  /*  Pointer to the file name of the object. */
PULONG            pcb;          /*  Size of the file-name buffer. */
BOOL              fQualified;   /*  Indicates whether or not to query fully qualified path. */
BOOL              rc;           /*  Success indicator. */

rc = _wpQueryRealName(somSelf, pszFilename,
       pcb, fQualified);
```

wpQueryRealName Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpQueryRealName Parameter - pszFilename

pszFilename (PSZ) - output
Pointer to the file name of the object.

The pointer to the buffer in which to place the real file name of the object.

wpQueryRealName Parameter - pcb

pcb (PULONG) - in/out
Size of the file-name buffer.

If *pszFilename* is set to NULL, the actual length of the file is returned.

wpQueryRealName Parameter - fQualified

fQualified (BOOL) - input
Indicates whether or not to query fully qualified path.

TRUE	Return the fully qualified file name.
FALSE	Return the unqualified file name.

wpQueryRealName Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryRealName - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

pszFilename (PSZ) - output
Pointer to the file name of the object.

The pointer to the buffer in which to place the real file name of the object.

pcb (PULONG) - in/out
Size of the file-name buffer.

If *pszFilename* is set to NULL, the actual length of the file is returned.

fQualified (BOOL) - input
Indicates whether or not to query fully qualified path.

TRUE	Return the fully qualified file name.
FALSE	Return the unqualified file name.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryRealName - Remarks

This method returns the fully qualified file name for this object. Generally, the object's real name and an object's title are the same. For file systems that don't support the features of a title, for example, characters, mixed case, and spaces, the title is stored in the .LONGNAME extended attribute and then the title and real name may differ. The real name of the file object can be used with any of the DOSXXX functions that act on file names.

wpQueryRealName - Usage

This method can be called at any time in order to determine the physical file name for this object.

wpQueryRealName - How to Override

This method is generally not overridden.

wpQueryRealName - Related Methods

Related Methods

- wpQueryType
- wpSetRealName
- wpSetTitle

wpQueryRealName - Example Code

This example overrides _wpSetup, specifies setup strings, does some initialization, and writes some default data to the object's real filename. This will provide something to look at if the user drag/drops on an editor or selects the open/editor view.

```
SOM_Scope BOOL   SOMLINK myf_wpSetup(MYFILE *somSelf,
                 PSZ pszSetupString)
{
    MYFILEData *somThis = MYFILEGetData(somSelf);
    ULONG  cbBytesWritten;            /* Number of bytes written */
    APIRET rc;                        /* Return code */
    BOOL   fSuccess;                  /* Success flag */
    HFILE  hf;                        /* File handle */
    ULONG  ulAction;                  /* Action taken by DosOpen */
    CHAR   szObjectFilename[CCHMAXPATH];    /* Buffer for wpQueryRealName() */
    ULONG  cb  = sizeof(szObjectFilename);  /* Size of object */
    PSZ    pszDefaultText;            /* Default text */
    BOOL   rcParentCall;              /* Result of parent's method */
    CHAR   szValue[CCHMAXPATH+1];
    ULONG  cbBuffer;
    MYFILEMethodDebug("MYFILE","myf_wpSetup");

   /*   When the object is created from scratch, put some default
        text into the file on the hard disk                          */

   fSuccess =
   _wpQueryRealName(                /* query full-pathname of object's file */
      somSelf,                                 /* pointer to this object */
      szObjectFilename,                                /* return buffer */
      &cb,                                             /* sizeof buffer */
      TRUE);                          /* request fully qualified pathname? */


   if (fSuccess)
   {
      rc = DosOpen(szObjectFilename, &hf, &ulAction, 20L, FILE_NORMAL,
                   FILE_OPEN | FILE_CREATE,
                   OPEN_ACCESS_READWRITE | OPEN_SHARE_DENYNONE,
                   NULL);

      if (rc == NO_ERROR) {

         pszDefaultText = _clsQueryDefaultFileText(_MYFILE);
         DosWrite(hf, pszDefaultText, strlen(pszDefaultText), &cbBytesWritten);
         DosClose(hf);
      } /* endif */


   } else {
      _wpclsSetError( somSelf, MYFILE_QRN_FAILED );
   } /* endif (fSuccess) */

   rcParentCall = parent_wpSetup(somSelf,pszSetupString);

            /* Process setup strings we understand */

   cbBuffer = CCHMAXPATH;
   if ( _wpScanSetupString(somSelf, pszSetupString, "SOUND",
                           szValue,&cbBuffer) )
   {
      if ((szValue[0] == 'Y') && (szValue[1] == 'E') && (szValue[2] == 'S')) {
         _soundAlarm(somSelf);
      } /* endif */
   } /* endif */

  return( rcParentCall );
}
```

wpQueryRealName - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods

wpQueryRefreshFlags

wpQueryRefreshFlags - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method queries the refresh flags for a file system object. The refresh flags consist of a DIRTYBIT and a FOUNDBIT that are used to allow refresh to detect deleted files and new files after asking a folder to re-populate.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG             rc;       /*  Refresh flags. */

rc = _wpQueryRefreshFlags(somSelf);
```

wpQueryRefreshFlags Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpQueryRefreshFlags Return Value - rc

rc (ULONG) - returns
Refresh flags.

DIRTYBIT	(0x80000000) Used for refreshing the views of pre-existing objects.
FOUNDBIT	(0x40000000) Used for refreshing the views of new and deleted objects.

wpQueryRefreshFlags - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

rc (ULONG) - returns
Refresh flags.

DIRTYBIT	(0x80000000) Used for refreshing the views of pre-existing objects.
FOUNDBIT	(0x40000000) Used for refreshing the views of new and deleted objects.

wpQueryRefreshFlags - Remarks

The refresh flags consist of a DIRTYBIT and a FOUNDBIT that are used to detect deleted files and new files when refreshing the contents of a folder.

For example, if you develop a subclass of wpFolder which contains objects that represent entries in a database (where the database may get updated independently from your objects), you might want to resynchronize your folder full of objects with the contents of the database. One possible technique follows:

1. Loop through all of the objects in the folder and turn on the DIRTYBIT and turn off the FOUNDBIT for all of your objects.
2. Loop through the database. For every entry in the database, find the corresponding object.
 a. If the object exists, turn on the FOUNDBIT for the object.
 b. If the object doesn't exist, create a new object with the FOUNDBIT turned on and the DIRTYBIT turned off.
3. Loop through the objects in the folder again. For any object that has the FOUNDBIT turned off, delete the object (since there is no longer a corresponding entry in the database). For any object that has the DIRTYBIT turned on, update the view with the current contents of the object and turn its DIRTYBIT off.

This kind of technique can be implemented using private flags, or by using the flags provided by the wpQueryRefreshFlags and wpSetRefreshFlags methods.

wpQueryRefreshFlags - Usage

This method can be called at any time to retrieve the refresh flags for a file system object.

wpQueryRefreshFlags - How to Override

This method is generally not overridden.

wpQueryRefreshFlags - Related Methods

Related Methods

- wpSetRefreshFlags

wpQueryRefreshFlags - Example Code

This example shows how you can use the refresh flags to synchronize the contents of a folder with the contents of a data base. somSelf is the pointer to the folder containing the objects corresponding to the items in the database.

```
MyObject    *Object,*NextObject;
```

```
/* Mark all existing objects as not found   */
Object = (MyObject*)_wpQueryContent(somSelf,NULL,QC_FIRST);
while (Object)
{
   _wpSetRefreshFlags(Object,0);
   Object = (MyObject*)_wpQueryContent(somSelf,Object,QC_NEXT);
}

/* Loop through the data base. For each item in the data base, find the
 * corresponding object and call the wpSetRefreshFlags method for the
 * object to set FOUNDBIT on. For any item that does not have a
 * corresponding object, create the object and call the wpSetRefreshFlags
 * method for the new object to set the FOUNDBIT on.  */

/***** The code for the above described loop goes here *****/

/* Remove all objects corresponding to data base entries that have been
 * deleted from the data base  */
Object = (MyObject*)_wpQueryContent(somSelf,NULL,QC_FIRST);
while (Object)
{
   NextObject = (MyObject*)_wpQueryContent(somSelf,Object,QC_NEXT);
   ulFlags = _wpQueryRefreshFlags(Object);
   if (!(ulFlags & FOUNDBIT))
   {
      Object->wpDelete(0);
   }
   Object = NextObject;
}
```

wpQueryRefreshFlags - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryRefreshRate

wpQueryRefreshRate - Syntax

This method is specific to version 2.1, or higher, of the OS/2 operating system.

This instance method queries the current status-window update rate used by the power object when the automatic refresh option is enabled.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPower     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG        rc;       /*  Refresh rate value defined in minutes. */
```

```
rc = _wpQueryRefreshRate(somSelf);
```

wpQueryRefreshRate Parameter - somSelf

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

wpQueryRefreshRate Return Value - rc

rc (ULONG) - returns
Refresh rate value defined in minutes.

wpQueryRefreshRate - Parameters

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

rc (ULONG) - returns
Refresh rate value defined in minutes.

wpQueryRefreshRate - Related Methods

Related Methods

- wpAddPowerPage
- wpAddPowerViewPage
- wpChangePowerState
- wpQueryAutoRefresh
- wpQueryPowerConfirmation
- wpQueryPowerManagement
- wpQueryRefreshRate
- wpSetAutoRefresh
- wpSetPowerConfirmation
- wpSetPowerManagement
- wpSetRefreshRate

wpQueryRefreshRate - Topics

Select an item:
Syntax
Parameters
Returns
Related Methods
Glossary

wpQueryRemoteOptions

wpQueryRemoteOptions - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the printer object's remote options.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPrinter     *somSelf;             /*  Pointer to the object on which the method is being invoked. */
PULONG         pulRefreshInterval;  /*  Pointer to the time interval indicating when the printer object is refre
PULONG         pflAllJobs;          /*  Pointer to the flag indicating which jobs to display. */
BOOL           rc;                  /*  Success indicator. */

rc = _wpQueryRemoteOptions(somSelf, pulRefreshInterval,
       pflAllJobs);
```

wpQueryRemoteOptions Parameter - somSelf

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

wpQueryRemoteOptions Parameter - pulRefreshInterval

pulRefreshInterval (PULONG) - output
Pointer to the time interval indicating when the printer object is refreshed.

The time interval is expressed in seconds.

wpQueryRemoteOptions Parameter - pflAllJobs

pflAllJobs (PULONG) - output
Pointer to the flag indicating which jobs to display.

TRUE
Display all jobs waiting in the network queue.
FALSE
Display only the current user's jobs waiting in the network queue.

wpQueryRemoteOptions Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpQueryRemoteOptions - Parameters

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

pulRefreshInterval (PULONG) - output
Pointer to the time interval indicating when the printer object is refreshed.

The time interval is expressed in seconds.

pflAllJobs (PULONG) - output
Pointer to the flag indicating which jobs to display.

TRUE
Display all jobs waiting in the network queue.
FALSE
Display only the current user's jobs waiting in the network queue.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpQueryRemoteOptions - Usage

This method can be called at any time.

wpQueryRemoteOptions - How to Override

This method is generally not overridden.

wpQueryRemoteOptions - Related Methods

Related Methods

- wpSetRemoteOptions

wpQueryRemoteOptions - Example Code

```
BOOL        rc;
ULONG       ulOrigRefreshSeconds = 0;
ULONG       ulOrigAllJobsDisplay = 0;

ULONG       ulRefreshSeconds = 5;        /* Refresh interval of 5 seconds */
ULONG       ulAllJobsDisplay = TRUE;     /* Display all jobs              */

rc = _wpQueryRemoteOptions( somSelf
                          , &ulOrigRefreshSeconds
                          , &ulOrigAllJobsDisplay);

somPrintf("Original Refresh = %u, Jobs Display = %u.\n",
           ulOrigRefreshSeconds,ulOrigAllJobsDisplay);

rc = _wpSetRemoteOptions( somSelf
                        , ulRefreshSeconds
                        , ulAllJobsDisplay);
```

wpQueryRemoteOptions - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryRootFolder

wpQueryRootFolder - Syntax

This instance method returns the root folder object for the logical drive that is represented by the WPDisk object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDisk           *somSelf;     /*  Pointer to the object on which the method is being invoked. */
WPRootFolder     *RootFolder;  /*  Root folder object. */

RootFolder = _wpQueryRootFolder(somSelf);
```

wpQueryRootFolder Parameter - somSelf

somSelf (WPDisk *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDisk.

wpQueryRootFolder Return Value - RootFolder

RootFolder (WPRootFolder *) - returns
Root folder object.

Null if an error occurred.

wpQueryRootFolder - Parameters

somSelf (WPDisk *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDisk.

RootFolder (WPRootFolder *) - returns
Root folder object.

Null if an error occurred.

wpQueryRootFolder - Remarks

Every instance of the WPDisk class that is created in the system points to a root folder, which is the root directory of the logical device that the disk object represents.

wpQueryRootFolder - Usage

This method can be called at any time.

wpQueryRootFolder - How to Override

This method should not be overridden.

wpQueryRootFolder - Example Code

```
 ULONG           ulLogicalDriveNum = 0;   /* Logical drive number returned */
 WPRootFolder   *RootFolder;              /* Root folder for drive         */

  /*
   *  Query logical drive number for the WPDisk object
   */

   ulLogicalDriveNum = _wpQueryLogicalDrive( self );


   if (LogicalDrive == 0) {

     somPrintf("_wpQueryLogicalDrive failed");
     return 1;

   } /* endif */


figseg fit=12.
  /*
   *  Query root folder also
   */

   RootFolder = _wpQueryRootFolder( self );

   if (RootFolder == NULL) {

     somPrintf("_wpQueryRootFolder failed");
     return 1;

   } /* endif */
```

wpQueryRootFolder - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Glossary

wpQueryScreenGroupID (Intel)

wpQueryScreenGroupID (Intel) - Syntax

This method is specific to version 3 of the OS/2 operating system.

This instance method returns the screen group ID of the running application for a given object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
USHORT        usPrevSgId;  /*  Previous screen group ID. */
USHORT        usScreenId;  /*  Screen group ID. */

usScreenId = _wpQueryScreenGroupID(somSelf,
               usPrevSgId);
```

wpQueryScreenGroupID (Intel) Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryScreenGroupID (Intel) Parameter - usPrevSgId

usPrevSgId (USHORT) - input
Previous screen group ID.

0
The first ID is returned.

other
The next ID after the *usPrevSgId* is returned.

wpQueryScreenGroupID (Intel) Return Value - usScreenId

usScreenId (USHORT) - returns
Screen group ID.

wpQueryScreenGroupID (Intel) - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

usPrevSgId (USHORT) - input
Previous screen group ID.

0
The first ID is returned.
other
The next ID after the *usPrevSgId* is returned.

usScreenId (USHORT) - returns
Screen group ID.

wpQueryScreenGroupID (Intel) - Usage

This method can be called at any time.

wpQueryScreenGroupID (Intel) - How to Override

This method should not be overridden.

wpQueryScreenGroupID (Intel) - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Glossary

wpQueryShadowedObject

wpQueryShadowedObject - Syntax

This instance method is called to allow the shadow object to query the object it is currently linked to.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPShadow     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL          fLock;    /*  Lock object flag. */
WPObject     *Success;  /*  Success indicator. */

Success = _wpQueryShadowedObject(somSelf,
            fLock);
```

wpQueryShadowedObject Parameter - somSelf

somSelf (WPShadow *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPShadow.

wpQueryShadowedObject Parameter - fLock

fLock (BOOL) - input
Lock object flag.

If this flag is false, the newly created object will be made dormant whenever the object and the folder containing the object are closed.
If this flag is true, the new flag will remain awake until the caller issues the wpUnlockObject method on it.

wpQueryShadowedObject Return Value - Success

Success (WPObject *) - returns
Success indicator.

NULL	Error occurred
Other	A pointer to the object this shadow is linked to.

wpQueryShadowedObject - Parameters

somSelf (WPShadow *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPShadow.

fLock (BOOL) - input
Lock object flag.

If this flag is false, the newly created object will be made dormant whenever the object and the folder containing the object are closed.
If this flag is true, the new flag will remain awake until the caller issues the wpUnlockObject method on it.

Success (WPObject *) - returns
Success indicator.

NULL	Error occurred
Other	A pointer to the object this shadow is linked to.

wpQueryShadowedObject - Usage

This method is called at any time in order to determine the object that this shadow is currently linked to.

wpQueryShadowedObject - How to Override

This method is generally not overridden.

wpQueryShadowedObject - Related Methods

Related Methods

- wpCreateShadowObject

wpQueryShadowedObject - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpQueryShadowTextColor

wpQueryShadowTextColor - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method returns the current color being used for the shadow text in the specified view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         ulView;   /*  Specifies which view to query. */
ULONG         rv;       /*  The RGB color value of the current color being used for the icon text background. */

rv = _wpQueryShadowTextColor(somSelf, ulView);
```

wpQueryShadowTextColor Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpQueryShadowTextColor Parameter - ulView

ulView (ULONG) - input
Specifies which view to query.

OPEN_CONTENTS
Opens the Contents view.
OPEN_DEFAULT
Opens the default view (same as double-clicking).
OPEN_DETAILS
Opens the Details view.
OPEN_HELP
Displays the Help panel.
OPEN_RUNNING
Executes the object.
OPEN_SETTINGS
Opens the Settings notebook.
OPEN_TREE
Opens the Tree view.
OPEN_USER
Class-specific views have a greater value than this.

wpQueryShadowTextColor Return Value - rv

rv (ULONG) - returns
The RGB color value of the current color being used for the icon text background.

The RGB value is presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

wpQueryShadowTextColor - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulView (ULONG) - input
Specifies which view to query.

OPEN_CONTENTS
Opens the Contents view.
OPEN_DEFAULT
Opens the default view (same as double-clicking).
OPEN_DETAILS
Opens the Details view.
OPEN_HELP
Displays the Help panel.
OPEN_RUNNING
Executes the object.
OPEN_SETTINGS
Opens the Settings notebook.
OPEN_TREE
Opens the Tree view.
OPEN_USER
Class-specific views have a greater value than this.

rv (ULONG) - returns
The RGB color value of the current color being used for the icon text background.

The RGB value is presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

wpQueryShadowTextColor - Usage

This method can be called at any time in order to determine the color used to display the title of a shadow icon in a view of the folder.

wpQueryShadowTextColor - How to Override

This method is generally not overridden.

wpQueryShadowTextColor - Related Methods

Related Methods

- wpQueryIconTextBackgroundColor
- wpQueryIconTextColor
- wpQueryIconTextVisibility
- wpSetIconTextBackgroundColor
- wpSetIconTextColor
- wpSetIconTextVisibility
- wpSetShadowTextColor

wpQueryShadowTextColor - Example Code

This example lets you find the shadow text color for Icon View of the current folder.

```
ULONG        ulTextColor;

ulTextColor = _wpQueryShadowTextColor(somSelf,OPEN_CONTENTS);
/* ulTextColor is an encoded RGB color value */
```

wpQueryShadowTextColor - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQuerySrvrIdentity

wpQuerySrvrIdentity - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the fully qualified server name of this network group.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPServer     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PSZ           pszName;  /*  Fully qualified server name of the server group. */

pszName = _wpQuerySrvrIdentity(somSelf);
```

wpQuerySrvrIdentity Parameter - somSelf

somSelf (WPServer *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPServer.

wpQuerySrvrIdentity Return Value - pszName

pszName (PSZ) - returns
Fully qualified server name of the server group.

For example, LSServer.

wpQuerySrvrIdentity - Parameters

somSelf (WPServer *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPServer.

pszName (PSZ) - returns
Fully qualified server name of the server group.

For example, LSServer.

wpQuerySrvrIdentity - How to Override

This method is generally not overridden.

wpQuerySrvrIdentity - Related Methods

Related Methods

- wpQueryNetIdentity
- wpQueryObjectNetId
- wpSetObjectNetId

wpQuerySrvrIdentity - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpQueryStyle

wpQueryStyle - Syntax

This instance method allows the object to query its current class style.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
ULONG         ulnwQrySt;  /*  Object styles. */

ulnwQrySt = _wpQueryStyle(somSelf);
```

wpQueryStyle Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryStyle Return Value - ulnwQrySt

ulnwQrySt (ULONG) - returns
Object styles.

The flags that are *ored* together to indicate the object's style.

OBJSTYLE_CUSTOMICON
Icon is destroyed when object goes dormant.
OBJSTYLE_NOCOPY
This object cannot be copied.
OBJSTYLE_NODELETE
This object cannot be deleted.
OBJSTYLE_NODRAG
This object cannot be dragged.
OBJSTYLE_NODROPON
No other object can be dropped on this object; however, this object can be dropped on other objects.
OBJSTYLE_NOLINK
This object cannot have shadow created.
OBJSTYLE_NOMOVE
This object cannot move.
OBJSTYLE_NOPRINT
This object cannot be printed.
OBJSTYLE_NORENAME
This object cannot be renamed.
OBJSTYLE_NOSETTINGS
This object cannot be set.
OBJSTYLE_NOTVISIBLE
This object is hidden.
OBJSTYLE_TEMPLATE
This object is a template.
OBJSTYLE_LOCKEDINPLACE
Lock this object in place.

wpQueryStyle - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulnwQrySt (ULONG) - returns
Object styles.

The flags that are *ored* together to indicate the object's style.

OBJSTYLE_CUSTOMICON
Icon is destroyed when object goes dormant.
OBJSTYLE_NOCOPY
This object cannot be copied.
OBJSTYLE_NODELETE
This object cannot be deleted.
OBJSTYLE_NODRAG
This object cannot be dragged.
OBJSTYLE_NODROPON
No other object can be dropped on this object; however, this object can be dropped on other objects.
OBJSTYLE_NOLINK
This object cannot have shadow created.
OBJSTYLE_NOMOVE
This object cannot move.
OBJSTYLE_NOPRINT
This object cannot be printed.
OBJSTYLE_NORENAME
This object cannot be renamed.
OBJSTYLE_NOSETTINGS
This object cannot be set.
OBJSTYLE_NOTVISIBLE
This object is hidden.
OBJSTYLE_TEMPLATE
This object is a template.
OBJSTYLE_LOCKEDINPLACE
Lock this object in place.

wpQueryStyle - Usage

This method can be called at any time in order to determine the current style for an object class. This is the dynamic value of the style based on the current state of the object.

To determine the default style for an object class, the wpclsQueryStyle method should be called.

wpQueryStyle - How to Override

This method is intended to be overridden by object classes. The override of this method may return different style flags based on the current state of the object.

wpQueryStyle - Related Methods

Related Methods

- wpclsQueryStyle
- wpQueryStyle
- wpSetStyle

wpQueryStyle - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpQueryTitle

wpQueryTitle - Syntax

This instance method is called to allow the object to query its current title.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PSZ           pQTitle;  /*  Pointer to the object's title. */

pQTitle = _wpQueryTitle(somSelf);
```

wpQueryTitle Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryTitle Return Value - pQTitle

pQTitle (PSZ) - returns
Pointer to the object's title.

wpQueryTitle - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pQTitle (PSZ) - returns
Pointer to the object's title.

wpQueryTitle - Remarks

The object's title may be altered by the user at any time. Objects should always use this method to access the current title and never store the string pointer that is returned.

wpQueryTitle - Usage

This method can be called at any time in order to determine the current title of an object. To determine the default title for an object's class, the wpclsQueryTitle method should be called.

wpQueryTitle - How to Override

This method is generally not overridden.

wpQueryTitle - Related Methods

Related Methods

- wpQueryTitle
- wpclsQueryTitle
- wpSetTitle

wpQueryTitle - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override

Usage
Related Methods
Glossary

wpQueryTrueStyle

wpQueryTrueStyle - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method allows the object to query its current class true style.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
ULONG         ulnwQrySt;  /*  Object styles. */

ulnwQrySt = _wpQueryTrueStyle(somSelf);
```

wpQueryTrueStyle Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpQueryTrueStyle Return Value - ulnwQrySt

ulnwQrySt (ULONG) - returns
Object styles.

The flags that are *ored* together to indicate the object's style.

OBJSTYLE_CUSTOMICON
The icon is destroyed when the object goes dormant.
OBJSTYLE_NOCOPY
This object cannot be copied.
OBJSTYLE_NODELETE
This object cannot be deleted.
OBJSTYLE_NODRAG
This object cannot be dragged.
OBJSTYLE_NODROPON
No other object can be dropped on this object; however, this object can be dropped on other objects.
OBJSTYLE_NOLINK
This object cannot have a shadow created.
OBJSTYLE_NOMOVE

This object cannot be moved.
OBJSTYLE_NOPRINT
This object cannot be printed.
OBJSTYLE_NORENAME
This object cannot be renamed.
OBJSTYLE_NOSETTINGS
This object cannot be set.
OBJSTYLE_NOTVISIBLE
This object is hidden.
OBJSTYLE_TEMPLATE
This object is a template.

wpQueryTrueStyle - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulnwQrySt (ULONG) - returns
Object styles.

The flags that are *ored* together to indicate the object's style.

OBJSTYLE_CUSTOMICON
The icon is destroyed when the object goes dormant.
OBJSTYLE_NOCOPY
This object cannot be copied.
OBJSTYLE_NODELETE
This object cannot be deleted.
OBJSTYLE_NODRAG
This object cannot be dragged.
OBJSTYLE_NODROPON
No other object can be dropped on this object; however, this object can be dropped on other objects.
OBJSTYLE_NOLINK
This object cannot have a shadow created.
OBJSTYLE_NOMOVE
This object cannot be moved.
OBJSTYLE_NOPRINT
This object cannot be printed.
OBJSTYLE_NORENAME
This object cannot be renamed.
OBJSTYLE_NOSETTINGS
This object cannot be set.
OBJSTYLE_NOTVISIBLE
This object is hidden.
OBJSTYLE_TEMPLATE
This object is a template.

wpQueryTrueStyle - Remarks

The wpQueryTrueStyle method always returns the style flags for the object as last set by the wpModifyStyle or wpSetStyle methods. This represents the persistent style of the object.

The wpQueryStyle method also returns the style flags for the object, but this method can be overridden by a descendant class to reflect temporary or cosmetic changes to the object's style. For example, if an object can generally be moved but is temporarily not moveable, the wpQueryTrueStyle method will return a style that does not contain the OBJSTYLE_NOMOVE flag, while the wpQueryStyle method will return a style that contains the OBJSTYLE_NOMOVE flag.

wpQueryTrueStyle - Usage

This method can be called at any time in order to determine the current last set style for an object. This is the static definition of the object's style.

To determine the default style for an object class, the wpclsQueryStyle method should be called. This method can be called at any time in order to determine the

wpQueryTrueStyle - How to Override

This method is never overridden.

wpQueryTrueStyle - Related Methods

Related Methods

- wpModifyStyle
- wpQueryStyle
- wpclsQueryStyle
- wpSetStyle

wpQueryTrueStyle - Example Code

This example lets you find the true style for an object.

```
ULONG        ulTrueStyle;

ulTrueStyle = _wpQueryTrueStyle(somSelf);
```

wpQueryTrueStyle - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryType

wpQueryType - Syntax

This instance method is called to allow the object to query the type of its file.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PSZ               ptype;    /*  Pointer to a buffer containing the file type. */

ptype = _wpQueryType(somSelf);
```

wpQueryType Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpQueryType Return Value - ptype

ptype (PSZ) - returns
Pointer to a buffer containing the file type.

This string can contain a list of types delineated by a line-feed character; for example, "Plain Text\nC Code".

wpQueryType - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

ptype (PSZ) - returns
Pointer to a buffer containing the file type.

This string can contain a list of types delineated by a line-feed character; for example, "Plain Text\nC Code".

wpQueryType - Remarks

This method returns the type of a file-system-based object. The type of a file is designated by its .TYPE extended attribute value.

wpQueryType - Usage

This method can be called at any time in order to determine the type of the file object.

wpQueryType - How to Override

This method is generally not overridden.

wpQueryType - Related Methods

Related Methods

- wpAddFileTypePage
- wpSetTitle

wpQueryType - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpQueryWarnBeforeStart (WPProgram)

wpQueryWarnBeforeStart (WPProgram) - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method queries the program reference's Warn Before Start flag. If this flag is set, put up a warning dialog before starting the program.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgram     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
```

```
BOOL            rc;         /*  Success indicator */

rc = _wpQueryWarnBeforeStart(somSelf);
```

wpQueryWarnBeforeStart (WPProgram) Parameter - somSelf

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

wpQueryWarnBeforeStart (WPProgram) Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryWarnBeforeStart (WPProgram) - Parameters

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryWarnBeforeStart (WPProgram) - Usage

This method can be called at any time to obtin the Warn Before Start option for a program reference object.

wpQueryWarnBeforeStart (WPProgram) - How to Override

This method is not generally overridden.

wpQueryWarnBeforeStart (WPProgram) - Related Methods

Related Methods

- wpQueryWarnBeforeStart (WPProgramFile)
- wpSetWarnBeforeStart (WPProgram)
- wpSetWarnBeforeStart (WPProgramFile)

wpQueryWarnBeforeStart (WPProgram) - Example Code

This method obtains the Warn Before Start option for the current program reference object.

```
BOOL        bWarnBeforeStart;

bWarnBeforeStart = _wpQueryWarnBeforeStart(somSelf);
```

wpQueryWarnBeforeStart (WPProgram) - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpQueryWarnBeforeStart (WPProgramFile)

wpQueryWarnBeforeStart (WPProgramFile) - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method queries the program reference's Warn Before Start flag. If this flag is set, put up a warning dialog before starting the program.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgramFile     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL               rc;       /*  Success indicator */

rc = _wpQueryWarnBeforeStart(somSelf);
```

wpQueryWarnBeforeStart (WPProgramFile) Parameter - somSelf

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

wpQueryWarnBeforeStart (WPProgramFile) Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryWarnBeforeStart (WPProgramFile) - Parameters

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpQueryWarnBeforeStart (WPProgramFile) - Usage

This method can be called at any time to obtin the Warn Before Start option for a program reference object.

wpQueryWarnBeforeStart (WPProgramFile) - How to Override

This method is not generally overridden.

wpQueryWarnBeforeStart (WPProgramFile) - Related Methods

Related Methods

- wpQueryWarnBeforeStart (WPProgram)
- wpSetWarnBeforeStart (WPProgram)
- wpSetWarnBeforeStart (WPProgramFile)

wpQueryWarnBeforeStart (WPProgramFile) - Example Code

This method obtains the Warn Before Start option for the current program file object.

```
BOOL        bWarnBeforeStart;

bWarnBeforeStart = _wpQueryWarnBeforeStart(somSelf);
```

wpQueryWarnBeforeStart (WPProgramFile) - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpReadImageFile

wpReadImageFile - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method reads the image data for the image file and stores it in the instance data for the object. If the image data has already been read, this method simply returns TRUE. It does not refresh the image data.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPBitmap     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL32        rc;       /*  Success indicator. */

rc = _wpReadImageFile(somSelf);
```

wpReadImageFile Parameter - somSelf

somSelf (WPBitmap *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPBitmap.

wpReadImageFile Return Value - rc

rc (BOOL32) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpReadImageFile - Parameters

somSelf (WPBitmap *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPBitmap.

rc (BOOL32) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpReadImageFile - Remarks

In the implementation of the WPBitmap class in the Workplace Shell, the wpReadImageFile method reads the data from the image file, extracts the bitmap that most closely matches the characteristics of the display, and saves the data associated with that image. For any other image type, the wpReadImageFile method should read the data from the image file, select the appropriate image (if the file can contain images for multiple resolution displays), and convert the image data to bitmap format.

wpReadImageFile - Usage

This method reads the image data from the image data file and converts it to the format used for an OS/2 Bitmap. This method stores the data in a buffer pointed to by the instance data for the object.

Various WPImageFile methods call the wpReadImageFile method to read the data from an image file and convert it to a bitmap.

wpReadImageFile - How to Override

This method must be overridden by any subclass of WPImageFile.

wpReadImageFile - Related Methods

Related Methods

- wpQueryBitmapData
- wpQueryBitmapHandle
- wpQueryBitmapInfoHeader
- wpSetBitmapData
- wpWriteImageFile

wpReadImageFile - Example Code

This example shows a simplified version of the code in the WPBitmap object class. This example assumes that the image file only contains a single image and it is already in bitmap format.

```
SOM_Scope BOOL32 bmp_wpReadImageFile(WPBitmap *somSelf)
{
   WPBitmapData *somThis = WPBitmapGetData(somSelf);

   CHAR        szFileName[CCHMAXPATH];
   HFILE       hFile = NULLHANDLE;
   ULONG       ulAction;
   FILESTATUS3 FileStatus;
   ULONG       ulFileSize;
   PBYTE       pBitmapData = NULL;
   APIRET      RC;
   ULONG       ulBytesRead;
   PBITMAPFILEHEADER pbfh;

   /* If the data has already been read, just return    */
   if(_pBitmapData)
   {
      return TRUE;
   }

   /* Find the bitmap data file
    */
   _wpQueryFilename(somSelf, szFileName, TRUE);

   /* Open the file and find out how big it is   */
   if (DosOpen(szFileName,
               &hFile,
               &ulAction,
               0L,
               FILE_NORMAL,
               FILE_OPEN,
               OPEN_ACCESS_READONLY | OPEN_SHARE_DENYNONE,
               NULL))
   {
      return FALSE;
   }

   if (DosQueryFileInfo(hFile,
                        FIL_STANDARD,
                        &FileStatus,
                        sizeof(FileStatus)))
   {
      DosClose (hFile);
      return FALSE;
   }

   /* Allocate space for a buffer large enough to contain the entire file */
   ulFileSize = FileStatus.cbFile;
   pBitmapData = _wpAllocMem(somSelf,ulFileSize,FALSE);
   if (!pBitmapData)
   {
      DosClose (hFile);
      return FALSE;
   }
```

```
   /* Read the file into the buffer    */
   RC = DosRead (hFile, pBitmapData, ulFileSize, &ulBytesRead);
   DosClose(hFile);
   hFile = NULLHANDLE;
   if (RC || (ulBytesRead != ulFileSize))
   {
      _wpFreeMem (somSelf,pBitmapData);
      return FALSE;
   }

   /* Make sure the bitmap header has a valid type code and the header
    * fits within the file     */
   pbfh = (PBITMAPFILEHEADER) pBitmapData;
   if (pbfh->usType != BFT_BMAP ||
       (PCH)pbfh + pbfh->cbSize > pBitmapData + ulFileSize)
   {
      _wpFreeMem(somSelf,pBitmapData);
      return FALSE;
   }

   /* Save the bitmap data and exit    */
   _pBitmapData = pBitmapData;
   _ulBitmapDataSize = ulFileSize;
   return TRUE;
}
```

wpReadImageFile - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpRedrawCell

wpRedrawCell - Syntax

This instance method forces a palette object to repaint the specified cell area.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPalette     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PCELL          pCell;    /*  Pointer to the cell within the palette that needs repainting. */
BOOL           rc;       /*  Success indicator. */

rc = _wpRedrawCell(somSelf, pCell);
```

wpRedrawCell Parameter - somSelf

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

wpRedrawCell Parameter - pCell

pCell (PCELL) - input
Pointer to the cell within the palette that needs repainting.

wpRedrawCell Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	The method call was successful.
FALSE	The method call was unsuccessful.

wpRedrawCell - Parameters

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

pCell (PCELL) - input
Pointer to the cell within the palette that needs repainting.

rc (BOOL) - returns
Success indicator.

TRUE	The method call was successful.
FALSE	The method call was unsuccessful.

wpRedrawCell - Remarks

This method is used as cell values within the palette are altered. For example, when the color selector dialog is changing the color of a palette cell in the color palette, this method is invoked to refresh the color back in the open views of the color palette object.

wpRedrawCell - Usage

This method may be called at any time to force the palette object to repaint the specified cell in all of its currently open views.

wpRedrawCell - How to Override

Overriding this method is not recommended.

wpRedrawCell - Related Methods

Related Methods

- wpPaintCell

wpRedrawCell - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpRefresh

wpRefresh - Syntax

This instance method is called to force new file system information to be updated.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
ULONG             ulView;      /*  Flag indicating type of view being refreshed. */
PVOID             pReserved;   /*  Reserved; value must be NULL. */
BOOL              rc;          /*  Success indicator. */

rc = _wpRefresh(somSelf, ulView, pReserved);
```

wpRefresh Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpRefresh Parameter - ulView

ulView (ULONG) - input
Flag indicating type of view being refreshed.

This parameter takes one of the following flags or any other defined refresh view for that class.

OPEN_CONTENTS
Refresh contents (icon) view.
OPEN_DETAILS
Refresh details view.
OPEN_TREE
Refresh tree view.

wpRefresh Parameter - pReserved

pReserved (PVOID) - input
Reserved; value must be NULL.

wpRefresh Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpRefresh - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

ulView (ULONG) - input
Flag indicating type of view being refreshed.

This parameter takes one of the following flags or any other defined refresh view for that class.

OPEN_CONTENTS
Refresh contents (icon) view.
OPEN_DETAILS
Refresh details view.
OPEN_TREE
Refresh tree view.

pReserved (PVOID) - input
Reserved; value must be NULL.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpRefresh - Usage

This method can be called at any time to refresh the contents of a folder.

wpRefresh - How to Override

This method is generally not overridden.

wpRefresh - Related Methods

Related Methods

- wpPopulate

wpRefresh - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpRefreshDrawer

wpRefreshDrawer - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method updates the drawer after changes have been made to the set of objects in the Toolbar or drawer.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
ULONG            ulDrawer;  /*  Toolbar or drawer being refreshed. */

wpRefreshDrawer(somSelf, ulDrawer);
```

wpRefreshDrawer Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpRefreshDrawer Parameter - ulDrawer

ulDrawer (ULONG) - input
Toolbar or drawer being refreshed.

0
Toolbar.

>0
Drawer identifier.

wpRefreshDrawer - Return Value

There is no return value for this method.

wpRefreshDrawer - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

ulDrawer (ULONG) - input
Toolbar or drawer being refreshed.

0
Toolbar.
>0
Drawer identifier.

There is no return value for this method.

wpRefreshDrawer - How to Override

This method can be overridden to know when the drawer is being refreshed.

During the processing of wpRefreshDrawer, wpSetDrawerHWND is called to set *hwnd* to NULLHANDLE while the objects are being recreated. It then calls wpSetDrawerHWND to set *hwnd* back to handle of the frame window for the drawer.

wpRefreshDrawer - Related Methods

Related Methods

- wpQueryDrawerHWND
- wpSetDrawerHWND

wpRefreshDrawer - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpRegisterView

wpRegisterView - Syntax

This instance method is called to allow the object to register a new open view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
HWND          hwndFrame;     /*  Handle to the frame window containing the new view. */
PSZ           pszViewTitle;  /*  Pointer to a string containing the name of the view. */
BOOL          rc;            /*  Success indicator. */

rc = _wpRegisterView(somSelf, hwndFrame, pszViewTitle);
```

wpRegisterView Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpRegisterView Parameter - hwndFrame

hwndFrame (HWND) - input
Handle to the frame window containing the new view.

wpRegisterView Parameter - pszViewTitle

pszViewTitle (PSZ) - input
Pointer to a string containing the name of the view.

wpRegisterView Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpRegisterView - Parameters

somSelf (WPObject *) - input

Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndFrame (HWND) - input
Handle to the frame window containing the new view.

pszViewTitle (PSZ) - input
Pointer to a string containing the name of the view.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpRegisterView - Remarks

Registering a view sets the object title of the frame window to the object's title and adds a view title as the current view in the window list and title bar. In-use emphasis is managed by the wpAddToObjUseList method.

wpRegisterView - Usage

This method should be called anytime a new view of an object is created.

wpRegisterView - How to Override

This method is generally not overridden.

wpRegisterView - Related Methods

Related Methods

- wpAddToObjUseList
- wpOpen

wpRegisterView - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpReleaseJob

wpReleaseJob - Syntax

This instance method is called to release a job object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPJob     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL       rc;       /*  Success indicator. */

rc = _wpReleaseJob(somSelf);
```

wpReleaseJob Parameter - somSelf

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

wpReleaseJob Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpReleaseJob - Parameters

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpReleaseJob - How to Override

This method is generally not overridden.

wpReleaseJob - Related Methods

Related Methods

- wpHoldJob

wpReleaseJob - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpReleaseObjectMutexSem

wpReleaseObjectMutexSem - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

Each object has associated with it a mutex semaphore that can be used to serialize access to resources. This instance method is called to release the mutex semaphore for an object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         rc;       /*  Return code from DosReleaseMutexSem. */

rc = _wpReleaseObjectMutexSem(somSelf);
```

wpReleaseObjectMutexSem Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpReleaseObjectMutexSem Return Value - rc

rc (ULONG) - returns
Return code from DosReleaseMutexSem.

wpReleaseObjectMutexSem - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

rc (ULONG) - returns
Return code from DosReleaseMutexSem.

wpReleaseObjectMutexSem - Remarks

Each object has associated with it a mutex semaphore that can be used to serialize access to resources. This method is called to assert that the current thread indeed holds this mutex semaphore. Methods that expect to be invoked under semaphore protection can use this method to validate that the caller has indeed obtained ownership of the mutex semaphore.

This semaphore is used primarily to control access to the linked list of use items, but can be used for other purposes if needed. Folders also have a separate mutex semaphore, which is used to protect the content chain for that folder.

If you need both the object and the folder mutex semaphores, then you must obtain the folder mutex first.

If you need multiple object mutex semaphores, obtain them in object address order.

As with most semaphores, hold the semaphore for the minimum possible time.

wpReleaseObjectMutexSem - Usage

You can call the wpReleaseObjectMutexSem method at any time to allow other methods to obtain access to a resource.

wpReleaseObjectMutexSem - How to Override

This method is not generally overridden.

wpReleaseObjectMutexSem - Related Methods

Related Methods

- wpAssertObjectMutexSem
- wpRequestObjectMutexSem

wpReleaseObjectMutexSem - Example Code

In this example, you lock the folder pointed to by somSelf and enumerate all objects within the folder. When you are finished processing the objects in the folder, you unlock the folder.

```
WPObject    *Object;
BOOL        bSem;

bSem = !_wpRequestObjectMutexSem(somSelf,SEM_INDEFINITE_WAIT);
for (Object = _wpQueryContent(somSelf,NULL,QC_FIRST);
     Object;
     Object = _wpQueryContent(somSelf,Object,QC_NEXT));
{
   /***** Process the returned object *****/
}

if (bSem)
{
   _wpReleaseObjectMutexSem(somSelf);
}
```

wpReleaseObjectMutexSem - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpReleasePrinter

wpReleasePrinter - Syntax

This instance method is called to release a printer object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPrinter     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL           rc;       /*  Success indicator. */

rc = _wpReleasePrinter(somSelf);
```

wpReleasePrinter Parameter - somSelf

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

wpReleasePrinter Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpReleasePrinter - Parameters

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpReleasePrinter - How to Override

This method is generally not overridden.

wpReleasePrinter - Related Methods

Related Methods

- wpHoldPrinter

wpReleasePrinter - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpRender

wpRender - Syntax

This instance method is called to request a drag-rendering or drop-rendering format from the object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject        *somSelf;     /*  Pointer to the object on which the method is being invoked. */
PDRAGTRANSFER    pdxfer;      /*  Pointer to a DRAGTRANSFER structure. */
MRESULT          mresreturn;  /*  See the DM_RENDER message in the Presentation Manager Programming Reference fo

mresreturn = _wpRender(somSelf, pdxfer);
```

wpRender Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpRender Parameter - pdxfer

pdxfer (PDRAGTRANSFER) - input
Pointer to a DRAGTRANSFER structure.

wpRender Return Value - mresreturn

mresreturn (MRESULT) - returns
See the DM_RENDER message in the *Presentation Manager Programming Reference* for a description of the return value.

wpRender - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pdxfer (PDRAGTRANSFER) - input
Pointer to a DRAGTRANSFER structure.

mresreturn (MRESULT) - returns
See the DM_RENDER message in the *Presentation Manager Programming Reference* for a description of the return value.

wpRender - Remarks

The wpRender method is called when the object receives a DM_RENDER message. See the DM_RENDER message in the *Presentation Manager Programming Reference* for more information.

wpRender - Usage

This method is generally called only by the system as the folder containing the object processed by the DM_RENDER message.

wpRender - How to Override

This method should be overridden to return a class-specific rendering mechanism and format.

wpRender - Related Methods

Related Methods

- wpDragOver
- wpDrop
- wpEndConversation
- wpFormatDragItem

- wpRender
- wpRenderComplete

wpRender - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpRenderComplete

wpRenderComplete - Syntax

This instance method is called to notify the object that the drag or drop rendering request is complete.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject        *somSelf;     /*  Pointer to the object on which the method is being invoked. */
PDRAGTRANSFER    pdxfer;      /*  Pointer to a DRAGTRANSFER structure. */
ULONG            ulResult;    /*  Success indicator. */
MRESULT          mresreturn;  /*  Refer to the DM_RENDERCOMPLETE message in the Presentation Manager Programming

mresreturn = _wpRenderComplete(somSelf, pdxfer,
               ulResult);
```

wpRenderComplete Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpRenderComplete Parameter - pdxfer

pdxfer (PDRAGTRANSFER) - input
Pointer to a DRAGTRANSFER structure.

wpRenderComplete Parameter - ulResult

ulResult (ULONG) - input
Success indicator.

The flag indicating whether the operation was performed successfully. See DM_RENDERCOMPLETE in the *Presentation Manager Programming Reference* for more information about this parameter.

wpRenderComplete Return Value - mresreturn

mresreturn (MRESULT) - returns
Refer to the DM_RENDERCOMPLETE message in the *Presentation Manager Programming Reference* for a description of the return value.

wpRenderComplete - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pdxfer (PDRAGTRANSFER) - input
Pointer to a DRAGTRANSFER structure.

ulResult (ULONG) - input
Success indicator.

The flag indicating whether the operation was performed successfully. See DM_RENDERCOMPLETE in the *Presentation Manager Programming Reference* for more information about this parameter.

mresreturn (MRESULT) - returns
Refer to the DM_RENDERCOMPLETE message in the *Presentation Manager Programming Reference* for a description of the return value.

wpRenderComplete - Remarks

The wpRenderComplete method is called when the object receives a DM_RENDERCOMPLETE message. See the DM_RENDERCOMPLETE message in the *Presentation Manager Programming Reference* for more information.

wpRenderComplete - Usage

This method is generally called only by the system as the folder containing the object processed by the DM_RENDERCOMPLETE message.

wpRenderComplete - How to Override

This method should be overridden if the class has a special rendering mechanism and format.

wpRenderComplete - Related Methods

Related Methods

- wpDragOver
- wpDrop
- wpEndConversation
- wpFormatDragItem
- wpRender
- wpRenderComplete

wpRenderComplete - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpRenderFromClipboard

wpRenderFromClipboard - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method opens the clipboard and retrieves the type of data specified in the clipboard format passed in. It then writes the clipboard data to a file.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDataFile     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
ULONG           aRenderAs;  /*  An atom representing the clipboard format. */
BOOL            rc;         /*  Success indicator */
```

```
rc = _wpRenderFromClipboard(somSelf, aRenderAs);
```

wpRenderFromClipboard Parameter - somSelf

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

wpRenderFromClipboard Parameter - aRenderAs

aRenderAs (ULONG) - input
An atom representing the clipboard format.

See *Presentation Manager Programming Guide - Advanced Topics* for information on using atoms for defining clipboard formats.

wpRenderFromClipboard Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpRenderFromClipboard - Parameters

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

aRenderAs (ULONG) - input
An atom representing the clipboard format.

See *Presentation Manager Programming Guide - Advanced Topics* for information on using atoms for defining clipboard formats.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpRenderFromClipboard - Remarks

The clipboard format specified must be compatible with the type of data file being written.

wpRenderFromClipboard - Usage

This method can be called at any time to retrieve the contents of the clipboard and write it to a data file.

wpRenderFromClipboard - How to Override

This method is not generally overridden.

wpRenderFromClipboard - Related Methods

Related Methods

- wpclsAddClipboardAssoc
- wpclsEnumClipboardClasses

wpRenderFromClipboard - Example Code

This example gets a bitmap from the clipboard and writes it to the data file. If the clipboard does not contain a bitmap, the data file is not updated.

```
_wpRenderFromClipboard(BitmapFile,CF_BITMAP);
```

wpRenderFromClipboard - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpReplacementIsInEffect

wpReplacementIsInEffect - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method determines whether the given class is currently replaced by a specified class.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPClassManager     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
PSZ                 pszOldClass;  /*  Old class. */
PSZ                 pszNewClass;  /*  New class. */
BOOL                flReplaced;   /*  Flag indicating whether the given old class is currently replaced by the s

flReplaced = _wpReplacementIsInEffect(somSelf,
               pszOldClass, pszNewClass);
```

wpReplacementIsInEffect Parameter - somSelf

somSelf (WPClassManager *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class SOMClassMgr.

wpReplacementIsInEffect Parameter - pszOldClass

pszOldClass (PSZ) - input
Old class.

wpReplacementIsInEffect Parameter - pszNewClass

pszNewClass (PSZ) - input
New class.

wpReplacementIsInEffect Return Value - flReplaced

flReplaced (BOOL) - returns

Flag indicating whether the given old class is currently replaced by the specified new class.

TRUE
pszOldClass is currently replaced by *pszNewClass*
FALSE
pszOldClass is not currently replaced by *pszNewClass*

wpReplacementIsInEffect - Parameters

somSelf (WPClassManager *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class SOMClassMgr.

pszOldClass (PSZ) - input
Old class.

pszNewClass (PSZ) - input
New class.

flReplaced (BOOL) - returns
Flag indicating whether the given old class is currently replaced by the specified new class.

TRUE
pszOldClass is currently replaced by *pszNewClass*
FALSE
pszOldClass is not currently replaced by *pszNewClass*

wpReplacementIsInEffect - Usage

This method can be called at any time. It is invoked using SOMClasMgrObject.

wpReplacementIsInEffect - How to Override

This method is generally not overridden.

wpReplacementIsInEffect - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Glossary

wpReplaceObject

wpReplaceObject - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method replaces an object in another folder.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
WPObject     *targetObject;  /*  Object to replace. */
BOOL32        fMove;         /*  Type of move flag. */
BOOL32        rc;            /*  Success indicator */

rc = _wpReplaceObject(somSelf, targetObject,
       fMove);
```

wpReplaceObject Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpReplaceObject Parameter - targetObject

targetObject (WPObject *) - input
Object to replace.

wpReplaceObject Parameter - fMove

fMove (BOOL32) - input
Type of move flag.

TRUE	Move and replace object.
FALSE	Copy and replace object.

wpReplaceObject Return Value - rc

rc (BOOL32) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpReplaceObject - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

targetObject (WPObject *) - input
Object to replace.

fMove (BOOL32) - input
Type of move flag.

TRUE	Move and replace object.
FALSE	Copy and replace object.

rc (BOOL32) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpReplaceObject - Remarks

The wpReplaceObject method deletes the target object, copies the current object to the folder that contained the target object, then optionally deletes the current object (if the *fMove* parameter is set to TRUE).

wpReplaceObject - Usage

This method can be called at any time to replace one object with a copy of the current object.

wpReplaceObject - How to Override

This method is not generally overridden.

wpReplaceObject - Related Methods

Related Methods

- wpAppendObject

wpReplaceObject - Example Code

In this example, we replace the target object with a copy of the source object.

```
_wpReplaceObject(SourceObject,TargetObject,FALSE);
```

wpReplaceObject - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpRequestObjectMutexSem

wpRequestObjectMutexSem - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

Each object has associated with it a mutex semaphore that can be used to serialize access to resources. This instance method is called to request the mutex semaphore for an object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
ULONG         ulTimeout;   /*  Return code from WinRequestMutexSem. */
ULONG         rc;          /*  Success indicator */

rc = _wpRequestObjectMutexSem(somSelf, ulTimeout);
```

wpRequestObjectMutexSem Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpRequestObjectMutexSem Parameter - ulTimeout

ulTimeout (ULONG) - input
Return code from WinRequestMutexSem.

wpRequestObjectMutexSem Return Value - rc

rc (ULONG) - returns
Success indicator

TRUE	The object semaphore is held by the calling thread.
FALSE	The object semaphore is not held by the calling thread.

wpRequestObjectMutexSem - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulTimeout (ULONG) - input
Return code from WinRequestMutexSem.

rc (ULONG) - returns
Success indicator

TRUE	The object semaphore is held by the calling thread.
FALSE	The object semaphore is not held by the calling thread.

wpRequestObjectMutexSem - Remarks

The Workplace Shell calls the wpRequestObjectMutexSem method to stop other threads from modifying an object while it is updating it. For example, when the Workplace Shell is enumerating the objects within a folder, it calls the wpRequestObjectMutexSem method for the folder to stop new objects from being added to the folder or old objects from being removed from the folder.

Each object has associated with it a mutex semaphore that can be used to serialize access to resources. This method is called to assert that the current thread indeed holds this mutex semaphore. Methods that expect to be invoked under semaphore protection can use this method to validate that the caller has indeed obtained ownership of the mutex semaphore.

This semaphore is used primarily to control access to the linked list of use items, but can be used for other purposes if needed. Folders also

have a separate mutex semaphore, which is used to protect the content chain for that folder.

If you need both the object and the folder mutex semaphores, then you must obtain the folder mutex first.

If you need multiple object mutex semaphores, obtain them in object address order.

As with most semaphores, hold the semaphore for the minimum possible time.

wpRequestObjectMutexSem - Usage

You can call the wpRequestObjectMutexSem method at any time to serialize access to an object.

wpRequestObjectMutexSem - How to Override

This method is not generally overridden.

wpRequestObjectMutexSem - Related Methods

Related Methods

- wpAssertObjectMutexSem
- wpReleaseObjectMutexSem

wpRequestObjectMutexSem - Example Code

In this example, you lock the folder pointed to by somSelf and enumerate all objects within the folder. When you are finished processing the objects in the folder, you unlock the folder.

```
WPObject    *Object;
BOOL        bSem;

bSem = !_wpRequestObjectMutexSem(somSelf,SEM_INDEFINITE_WAIT);
for (Object = _wpQueryContent(somSelf,NULL,QC_FIRST);
     Object;
     Object = _wpQueryContent(somSelf,Object,QC_NEXT));
{
   /***** Process the returned object *****/
}

if (bSem)
{
   _wpReleaseObjectMutexSem(somSelf);
}
```

wpRequestObjectMutexSem - Topics

Select an item:
Syntax

Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpRestore

wpRestore - Syntax

This instance method is called to allow the object to restore its views from the hidden or minimized states.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL          rc;       /*  Success indicator. */

rc = _wpRestore(somSelf);
```

wpRestore Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpRestore Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpRestore - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpRestore - Remarks

This method is the inverse of the wpHide method.

wpRestore - Usage

This method can be called at any time in order to restore all views of this object from the hidden or minimized state.

wpRestore - How to Override

This method is generally not overridden.

wpRestore - Related Methods

Related Methods

- wpHide
- wpRestore

wpRestore - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpRestoreCellData

wpRestoreCellData - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method restores the data for the specified cell.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPalette     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
PCELL         *pCell;       /*  Pointer to the cell whose data is to be restored. */
ULONG          ulIndex;     /*  Cell index. */
ULONG          ulCellSize;  /*  Number of bytes to be allocated for the cell. */
BOOL           rc;          /*  Success indicator. */

rc = _wpRestoreCellData(somSelf, pCell, ulIndex,
       ulCellSize);
```

wpRestoreCellData Parameter - somSelf

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

wpRestoreCellData Parameter - pCell

pCell (PCELL *) - in/out
Pointer to the cell whose data is to be restored.

wpRestoreCellData Parameter - ulIndex

ulIndex (ULONG) - input
Cell index.

This is the number of the cell. It must be within the range 0 to xCellCount*yCellCount-1.

wpRestoreCellData Parameter - ulCellSize

ulCellSize (ULONG) - input
Number of bytes to be allocated for the cell.

If 0 is specified, the system allocates enough space to hold the saved data.

wpRestoreCellData Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpRestoreCellData - Parameters

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

pCell (PCELL *) - in/out
Pointer to the cell whose data is to be restored.

ulIndex (ULONG) - input
Cell index.

This is the number of the cell. It must be within the range 0 to xCellCount*yCellCount-1.

ulCellSize (ULONG) - input
Number of bytes to be allocated for the cell.

If 0 is specified, the system allocates enough space to hold the saved data.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpRestoreCellData - Usage

This method is called by the system while it is processing wpSaveState for WPPalette.

wpRestoreCellData - How to Override

This method must be overridden by all subclasses that wish to save and restore part of their cell data.

It should be overridden when special processing is to be done for subclassed palettes. This method must be overridden by all subclasses that

change the size of their cell data.

For example, the scheme palette saves the name of the background bitmap for the scheme.

wpRestoreCellData - Related Methods

Related Methods

- wpSaveCellData

wpRestoreCellData - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpRestoreData

wpRestoreData - Syntax

This instance method is called to allow the object to restore its binary instance data.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
PSZ           pszClass;  /*  Pointer to the class name. */
ULONG         ulKey;     /*  Class-defined identifier. */
PBYTE         pValue;    /*  Address of the data to be restored. */
PULONG        pcbValue;  /*  Size of the data block to be restored. */
BOOL          rc;        /*  Success indicator. */

rc = _wpRestoreData(somSelf, pszClass, ulKey,
       pValue, pcbValue);
```

wpRestoreData Parameter - somSelf

somSelf (WPObject *) - input

Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpRestoreData Parameter - pszClass

pszClass (PSZ) - input
Pointer to the class name.

A pointer to a zero-terminated string that contains any unique string. The class name is recommended but not enforced.

wpRestoreData Parameter - ulKey

ulKey (ULONG) - input
Class-defined identifier.

A class-defined identifier that correlates to a particular instance data variable.

wpRestoreData Parameter - pValue

pValue (PBYTE) - in/out
Address of the data to be restored.

wpRestoreData Parameter - pcbValue

pcbValue (PULONG) - in/out
Size of the data block to be restored.

If *pValue* is NULL, the actual size is returned in *pcbValue* .

wpRestoreData Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpRestoreData - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pszClass (PSZ) - input
Pointer to the class name.

A pointer to a zero-terminated string that contains any unique string. The class name is recommended but not enforced.

ulKey (ULONG) - input
Class-defined identifier.

A class-defined identifier that correlates to a particular instance data variable.

pValue (PBYTE) - in/out
Address of the data to be restored.

pcbValue (PULONG) - in/out
Size of the data block to be restored.

If *pValue* is NULL, the actual size is returned in *pcbValue* .

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpRestoreData - Remarks

This method restores data that was saved by a call to wpSaveData.

wpRestoreData - Usage

This method can be called only during the processing of the wpRestoreState method.

wpRestoreData - How to Override

This method is generally not overridden.

wpRestoreData - Related Methods

Related Methods

- wpRestoreData
- wpRestoreLong

- wpRestoreState
- wpRestoreString
- wpSaveData
- wpSaveDeferred
- wpSaveImmediate
- wpSaveLong
- wpSaveState
- wpSaveString

wpRestoreData - Example Code

In this example, wpRestoreState is overridden to restore data from the OS2.INI file.

```
SOM_Scope void   SOMLINK UserSetTitle(nbk *somSelf,
                                      PSZ pszNewTitle)
{
    nbkData *somThis = nbkGetData(somSelf);
    nbkMethodDebug("nbk","UserSetTitle");

   strcpy ( _title, pszNewTitle );
}
 *    Override of  wpRestoreState method to restore our data
 *    from the OS2.INI file.
 */

SOM_Scope BOOL   SOMLINK wpRestoreState(nbk   *somSelf,
                                        ULONG ulReserved)

{
    BOOL fSuccess;
    ULONG ulCount;
    CHAR buf[40];

    nbkData *somThis = nbkGetData(somSelf);
    nbkMethodDebug("nbk","wpRestoreState");

    ulCount = 40;                               /* Length of data buffer */
    fSuccess = _wpRestoreData ( somSelf         /* Our pointer */
                      , (PSZ) "nbk"             /* Name of class */
                      , WPNBK_KEY               /* Key of data requested */
                      , (PBYTE) &buf            /* Where to put data */
                      , &ulCount   );           /* Length of data returned */

    if ( !fSuccess )
    {
        /* first time the object is being restored */

        /* initialize string to NULL */
        strcpy ( buf, "" );

        /* Save initial string to OS2.INI file */
        _wpSaveData ( somSelf                /* Pointer to ourself */
                    , (PSZ) "nbk"            /* Class name */
                    , WPNBK_KEY              /* Key of data */
                    , (PBYTE) buf            /* Pointer to data */
                    , 40 );                  /* Length of data to write */
    } /* endif */

    _UserSetTitle ( somSelf, buf );

    return (parent_wpRestoreState(somSelf,ulReserved));
}
```

wpRestoreData - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpRestoreFldrRunObjs

wpRestoreFldrRunObjs - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method is called when a work area folder is opened. It goes through the list of objects last saved by wpStoreFldrRunObjs and reopens all views of those objects that were open when the work area folder was last closed.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL          rc;       /*  Success indicator */

rc = _wpRestoreFldrRunObjs(somSelf);
```

wpRestoreFldrRunObjs Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpRestoreFldrRunObjs Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpRestoreFldrRunObjs - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpRestoreFldrRunObjs - Remarks

The system calls this method when it is opening a work area folder. This method is only called on folders with the FOI_WORKAREA flag set when the folder is opened.

wpRestoreFldrRunObjs - Usage

This method can be called at any time to restore all objects that were open when a work area folder was last closed.

wpRestoreFldrRunObjs - How to Override

This method is not generally overridden.

wpRestoreFldrRunObjs - Related Methods

Related Methods

- wpHideFldrRunObjs
- wpStoreFldrRunObjs

wpRestoreFldrRunObjs - Example Code

This example restores all of the objects that were open when the specified folder was last closed.

```
_wpRestoreFldrRunObjs(WorkAreaFolder);
```

wpRestoreFldrRunObjs - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpRestoreLong

wpRestoreLong - Syntax

This instance method is called to allow the object to restore a 32-bit instance data value.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
PSZ           pszClass;  /*  Pointer to the class name. */
ULONG         ulKey;     /*  Class-defined identifier. */
PULONG        pulValue;  /*  The address of the long value. */
BOOL          rc;        /*  Success indicator. */

rc = _wpRestoreLong(somSelf, pszClass, ulKey,
       pulValue);
```

wpRestoreLong Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpRestoreLong Parameter - pszClass

pszClass (PSZ) - input
Pointer to the class name.

A pointer to a zero-terminated string which contains any unique string. The class name is recommended but not enforced.

wpRestoreLong Parameter - ulKey

ulKey (ULONG) - input
Class-defined identifier.

A class-defined identifier that correlates to a particular instance data variable.

wpRestoreLong Parameter - pulValue

pulValue (PULONG) - output
The address of the long value.

wpRestoreLong Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpRestoreLong - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pszClass (PSZ) - input
Pointer to the class name.

A pointer to a zero-terminated string which contains any unique string. The class name is recommended but not enforced.

ulKey (ULONG) - input
Class-defined identifier.

A class-defined identifier that correlates to a particular instance data variable.

pulValue (PULONG) - output
The address of the long value.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpRestoreLong - Remarks

This method restores a 32-bit data value that was saved by a call to wpSaveLong.

wpRestoreLong - Usage

This method can be called only during the processing of the wpRestoreState method.

wpRestoreLong - How to Override

This method is generally not overridden.

wpRestoreLong - Related Methods

Related Methods

- wpRestoreData
- wpRestoreLong
- wpRestoreState
- wpRestoreString
- wpSaveData
- wpSaveDeferred
- wpSaveImmediate
- wpSaveLong
- wpSaveState
- wpSaveString

wpRestoreLong - Example Code

This example demonstrates override of wpRestoreState to save and restore a ULONG.

```
SOM_Scope BOOL   SOMLINK wpRestoreState(LockBox *somSelf,
  ULONG ulReserved)
{
    ULONG  ulSecretCode;       /* Secret code */
    BOOL   fSuccess;           /* Success or failure flag */

    LockBoxData *somThis = LockBoxGetData(somSelf);
    LockBoxMethodDebug("LockBox","wpRestoreState");

    fSuccess = _wpRestoreLong ( somSelf       /* Pointer to thy self */
                      , "LockBox"             /* Class name */
                      , LOCKBOX_KEY_SECRET    /* Key of data to get */
                      , &ulSecretCode    );   /* Data returned here */

    if ( !fSuccess )
    {
        ulSecretCode = 0L;    /* On first restore, make secret code 0 */

        /* save initial code to OS2.INI file */

        _wpSaveLong ( somSelf               /* Pointer to me */
```

```
                    , "LockBox"             /* Class name */
                    , LOCKBOX_KEY_SECRET    /* Key of data */
                    , ulSecretCode );       /* Data */
    }
    return (parent_wpRestoreState(somSelf,ulReserved));
}
```

wpRestoreLong - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpRestoreState

wpRestoreState - Syntax

This method restores the state of the object which was saved during the processing of the wpSaveState method.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
ULONG         ulReserved;  /*  Reserved value; must be 0. */
BOOL          rc;          /*  Success indicator. */

rc = _wpRestoreState(somSelf, ulReserved);
```

wpRestoreState Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpRestoreState Parameter - ulReserved

ulReserved (ULONG) - input
Reserved value; must be 0.

wpRestoreState Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpRestoreState - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulReserved (ULONG) - input
Reserved value; must be 0.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpRestoreState - Remarks

This method restores the state of the object that was saved during the processing of the wpSaveState method.

wpRestoreState - Usage

This method is generally called only by the system while it is processing the wpInitData method.

wpRestoreState - How to Override

This method should be overridden by all classes that provide settings that can be saved. An override of the wpSaveState method is a prerequisite if persistent instance data is desired.

Override processing of this method typically includes a series of calls to any combination of the restore state methods:

- wpRestoreData
- wpRestoreLong
- wpRestoreString

wpRestoreState - Related Methods

Related Methods

- wpRestoreData
- wpRestoreLong
- wpRestoreState
- wpRestoreString
- wpSaveData
- wpSaveDeferred
- wpSaveImmediate
- wpSaveLong
- wpSaveState
- wpSaveString

wpRestoreState - Example Code

In this example, wpRestoreState is overridden to restore data from the OS2.INI file.

```
SOM_Scope void   SOMLINK UserSetTitle(nbk *somSelf,
                                      PSZ pszNewTitle)
{
    nbkData *somThis = nbkGetData(somSelf);
    nbkMethodDebug("nbk","UserSetTitle");

   strcpy ( _title, pszNewTitle );
}
/*
 *    Override of  wpRestoreState method to restore our data
 *    from the OS2.INI file.
 */

SOM_Scope BOOL   SOMLINK wpRestoreState(nbk   *somSelf,
                                        ULONG ulReserved)

{
    BOOL fSuccess;
    ULONG ulCount;
    CHAR buf[40];

    nbkData *somThis = nbkGetData(somSelf);
    nbkMethodDebug("nbk","wpRestoreState");

    ulCount = 40;                              /* Length of data buffer */
    fSuccess = _wpRestoreData ( somSelf        /* Our pointer */
                      , (PSZ) "nbk"            /* Name of class */
                      , WPNBK_KEY              /* Key of data requested */
                      , (PBYTE) &buf           /* Where to put data */
                      , &ulCount   );          /* Length of data returned */

    if ( !fSuccess )
    {
        /* first time the object is being restored */

        /* initialize string to NULL */
        strcpy ( buf, "" );

        /* Save initial string to OS2.INI file */
        _wpSaveData ( somSelf               /* Pointer to ourself */
                    , (PSZ) "nbk"           /* Class name */
                    , WPNBK_KEY             /* Key of data */
```

```
                    , (PBYTE) buf           /* Pointer to data */
                    , 40 );                 /* Length of data to write */
    } /* endif */

    _UserSetTitle ( somSelf, buf );

    return (parent_wpRestoreState(somSelf,ulReserved));
}
```

wpRestoreState - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpRestoreString

wpRestoreString - Syntax

This instance method is called to allow the object to restore an ASCIIZ instance data string.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
PSZ           pszClass;  /*  Pointer to the class name. */
ULONG         ulKey;     /*  Class-defined identifier. */
PSZ           pszValue;  /*  Address of the string to be restored. */
PULONG        pcbValue;  /*  Size of string to be restored. */
BOOL          rc;        /*  Success indicator. */

rc = _wpRestoreString(somSelf, pszClass, ulKey,
       pszValue, pcbValue);
```

wpRestoreString Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpRestoreString Parameter - pszClass

pszClass (PSZ) - input
Pointer to the class name.

A pointer to a zero-terminated string that contains any unique string. The class name is recommended but not enforced.

wpRestoreString Parameter - ulKey

ulKey (ULONG) - input
Class-defined identifier.

A class-defined identifier that correlates to a particular instance data variable.

wpRestoreString Parameter - pszValue

pszValue (PSZ) - in/out
Address of the string to be restored.

wpRestoreString Parameter - pcbValue

pcbValue (PULONG) - in/out
Size of string to be restored.

If *pszValue* is NULL, the actual size is returned in *pcbValue* .

wpRestoreString Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpRestoreString - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pszClass (PSZ) - input
Pointer to the class name.

A pointer to a zero-terminated string that contains any unique string. The class name is recommended but not enforced.

ulKey (ULONG) - input
Class-defined identifier.

A class-defined identifier that correlates to a particular instance data variable.

pszValue (PSZ) - in/out
Address of the string to be restored.

pcbValue (PULONG) - in/out
Size of string to be restored.

If *pszValue* is NULL, the actual size is returned in *pcbValue* .

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpRestoreString - Remarks

This method restores an ASCIIZ string that was saved by a call to wpSaveString.

wpRestoreString - Usage

This method can be called only during the processing of the wpRestoreState method.

wpRestoreString - How to Override

This method is generally not overridden.

wpRestoreString - Related Methods

Related Methods

- wpRestoreData
- wpRestoreLong
- wpRestoreState
- wpRestoreString
- wpSaveData
- wpSaveDeferred
- wpSaveImmediate

- wpSaveLong
- wpSaveState
- wpSaveString

wpRestoreString - Example Code

This example demonstrates an override of wpRestoreState method to save string data.

```
SOM_Scope BOOL   SOMLINK wpRestoreState(nbk   *somSelf,
                                        ULONG ulReserved)
{
    BOOL fSuccess;       /* Success flag */
    ULONG ulCount;       /* Length of title string */
    UCHAR uchName[40];   /* Buffer for title */

    nbkData *somThis = nbkGetData(somSelf);
    nbkMethodDebug("nbk","wpRestoreState");

    ulCount = 40L;                              /* Size of input buffer */
    fSuccess = _wpRestoreString ( somSelf       /* Self pointer */
                      , (PSZ) "nbk"             /* Class name */
                      , WPNBK_KEY               /* Key for data */
                      , &uchName                /* String to receive data */
                      , &ulCount   );           /* Amount of data returned */

    if ( !fSuccess ) {    /* First time object is being restored */

        // initialize string to NULL
        strcpy ( uchName, "" );

        // save initial string to ini file
        _wpSaveString ( somSelf                 /* Self pointer */
                     , (PSZ) "nbk"              /* Class name */
                     , WPNBK_KEY                /* Key of data */
                     , uchName     );           /* String to write */
    }

    _UserSetTitle ( somSelf, uchName );   /* Set an instance variable */

    return (parent_wpRestoreState(somSelf,ulReserved));
}
```

wpRestoreString - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpSaveCellData

wpSaveCellData - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method saves the data for the specified cell.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPalette     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PCELL          pCell;    /*  Pointer to the cell whose data is to be saved. */
ULONG          ulIndex;  /*  Cell index. */
BOOL           rc;       /*  Success indicator. */

rc = _wpSaveCellData(somSelf, pCell, ulIndex);
```

wpSaveCellData Parameter - somSelf

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

wpSaveCellData Parameter - pCell

pCell (PCELL) - input
Pointer to the cell whose data is to be saved.

wpSaveCellData Parameter - ulIndex

ulIndex (ULONG) - input
Cell index.

This is the number of the cell. It must be within the range 0 to xCellCount*yCellCount-1.

wpSaveCellData Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.

FALSE	Error occurred.

wpSaveCellData - Parameters

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

pCell (PCELL) - input
Pointer to the cell whose data is to be saved.

ulIndex (ULONG) - input
Cell index.

This is the number of the cell. It must be within the range 0 to xCellCount*yCellCount-1.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSaveCellData - Usage

This method is called by the system while it is processing wpSaveState for WPPalette.

wpSaveCellData - How to Override

This method should be overridden by all subclasses that wish to save and restore part of their cell data.

It should be overridden when special processing is to be done for subclassed palettes. This method must be overridden by all subclasses that change the size of their cell data.

For example, the scheme palette saves the name of the background bit map for the scheme.

wpSaveCellData - Related Methods

Related Methods

- wpRestoreCellData

wpSaveCellData - Topics

Select an item:
Syntax

Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpSaveData

wpSaveData - Syntax

This instance method is called to allow an object to save its binary instance data.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
PSZ           pszClass;  /*  Pointer to the class name. */
ULONG         ulKey;     /*  Class-defined identifier. */
PBYTE         pValue;    /*  Address of the block of data to be stored. */
ULONG         cbValue;   /*  Size of the block of data to be stored. */
BOOL          rc;        /*  Success indicator. */

rc = _wpSaveData(somSelf, pszClass, ulKey,
       pValue, cbValue);
```

wpSaveData Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSaveData Parameter - pszClass

pszClass (PSZ) - input
Pointer to the class name.

A pointer to a zero-terminated string that contains any unique string. The class name is recommended but not enforced.

wpSaveData Parameter - ulKey

ulKey (ULONG) - input
Class-defined identifier.

A class-defined identifier that correlates to a particular instance data variable.

wpSaveData Parameter - pValue

pValue (PBYTE) - input
Address of the block of data to be stored.

wpSaveData Parameter - cbValue

cbValue (ULONG) - input
Size of the block of data to be stored.

wpSaveData Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSaveData - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pszClass (PSZ) - input
Pointer to the class name.

A pointer to a zero-terminated string that contains any unique string. The class name is recommended but not enforced.

ulKey (ULONG) - input
Class-defined identifier.

A class-defined identifier that correlates to a particular instance data variable.

pValue (PBYTE) - input
Address of the block of data to be stored.

cbValue (ULONG) - input
Size of the block of data to be stored.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSaveData - Remarks

The saved data can be restored by issuing a call to wpRestoreData.

wpSaveData - Usage

This method can be called only during the processing of the wpSaveState method.

wpSaveData - How to Override

This method is generally not overridden.

wpSaveData - Related Methods

Related Methods

- wpRestoreData
- wpRestoreLong
- wpRestoreState
- wpRestoreString
- wpSaveData
- wpSaveDeferred
- wpSaveImmediate
- wpSaveLong
- wpSaveState
- wpSaveString

wpSaveData - Example Code

In this example, wpSaveState is overridden to save instance data from the OS2.INI file.

```
#define WPNBK_KEY        1

/*  Instance method to return title information */

SOM_Scope PSZ   SOMLINK UserGetTitle(nbk *somSelf)
{
    nbkData *somThis = nbkGetData(somSelf);
    nbkMethodDebug("nbk","UserGetTitle");

    return (PSZ) _title;
```

```
}

/* Override of wpSaveState to save our instance data in OS2.INI */

SOM_Scope BOOL   SOMLINK wpSaveState(nbk *somSelf)
{
    PSZ  psz;

    nbkData *somThis = nbkGetData(somSelf);
    nbkMethodDebug("nbk","wpSaveState");

    psz = _UserGetTitle ( somSelf );    /* Get data to be saved */

    /* Save our data to the OS2.INI file  */

     _wpSaveData ( somSelf              /* Pointer to us */
                 , "nbk"                /* Class name */
                 , WPNBK_KEY            /* Key for data */
                 , psz                  /* Pointer to data */
                 , 40 );                /* Length of data  */

    return (parent_wpSaveState(somSelf));  /* Let parent store data */
}
```

wpSaveData - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpSaveDeferred

wpSaveDeferred - Syntax

This instance method saves a given object asynchronously on a separate thread.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL          rc;       /*  Success indicator. */

rc = _wpSaveDeferred(somSelf);
```

wpSaveDeferred Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSaveDeferred Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSaveDeferred - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSaveDeferred - Remarks

This method is invoked by a Workplace Shell class each time a wpSetXXX method is used. It is recommended that wpSaveDeferred be used in most cases. The only exception is if the state data has changed in some critical way. For example, if the object wanted to save a new password, it would use wpSaveImmediate instead of wpSaveDeferred to guarantee that the password is saved before continuing.

Note: wpSaveDeferred can be called only after an object has been initialized. This method can be called from within wpObjectReady but not from wpRestoreState. To determine if an object is initialized, use wpIsObjectInitialized.

wpSaveDeferred - Usage

This method should be called periodically to get instance data written back to persistent storage when important state variables are altered.

wpSaveDeferred - How to Override

This method should not be overridden.

wpSaveDeferred - Related Methods

Related Methods

- wpSaveImmediate

wpSaveDeferred - Example Code

```
MRESULT EXPENTRY MyWindowProc   ( HWND hwndDlg, ULONG msg
                                , MPARAM mp1,   MPARAM mp2 )
{
    SOMAny  *somSelf;      /* Temporary ptr to object instance          */
    PWINDATA pwin;         /* Window data                               */
    PSZ      psz;          /* User title to manipulate                  */
    HWND     hwndEntry;    /* Window handle                             */
    CHAR     buf[30];      /* Buffer                                    */

    switch( msg )
    {
        case WM_INITDLG:

            somSelf = (SOMAny *) mp2;   /* obtain creation parms */

            pwin = (PWINDATA) _wpAllocMem ( somSelf, sizeof (WINDATA), NULL );

            WinSetWindowPtr (hwndDlg, QWL_USER, pwin );

            /* Initialize the windata structure */
            pwin->somSelf     = somSelf;
            pwin->somClassObj = _somGetClass ( pwin->somSelf );

            /* query title string from instance data, and fill  */
            /* entryfield with the string                       */

            psz = _UserGetTitle ( pwin->somSelf );

            /* Change title, and then save it to the OS2.INI file
               whenever it is convenient */

            strcat( psz, ":1" );

            _UserSetTitle ( pwin->somSelf, (PSZ) &buf );

            /* return TRUE to tell PM that we changed focus */
            return (MRESULT) TRUE ;

            _wpSaveDeferred ( pwin -> somSelf );

            hwndEntry = WinWindowFromID ( hwndDlg, DLG_ENTRYFIELD );

            WinSetWindowText ( hwndEntry, psz );

            WinSetFocus      ( HWND_DESKTOP, hwndEntry );
```

wpSaveDeferred - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpSaveImmediate

wpSaveImmediate - Syntax

This instance method is called to save the current state of an object synchronously.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL          rc;       /*  Success indicator. */

rc = _wpSaveImmediate(somSelf);
```

wpSaveImmediate Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSaveImmediate Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSaveImmediate - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSaveImmediate - Remarks

This method is called automatically for all objects that are flagged as needing to save their data by wpSaveDeferred; however, an object can call this method on itself at any time when a critical instance variable is changed. It is recommended that wpSaveDeferred be used in most cases. If you do choose to use this method, you should not call it from your user-interface thread; instead, you should use a background thread.

This method will cause the wpSaveState method to be called.

wpSaveImmediate - How to Override

This method is generally not overridden.

wpSaveImmediate - Related Methods

Related Methods

- wpRestoreData
- wpRestoreLong
- wpRestoreState
- wpRestoreString
- wpSaveData
- wpSaveDeferred
- wpSaveImmediate
- wpSaveLong
- wpSaveState
- wpSaveString

wpSaveImmediate - Example Code

```
MRESULT EXPENTRY MyDialogProc   ( HWND hwndDlg, ULONG msg
                                , MPARAM mp1,   MPARAM mp2 )
{
    SOMAny  *somSelf;      /* Temporary ptr to object instance          */
    PWINDATA pwin;         /* Window data                               */
    PSZ      psz;          /* User title to manipulate                  */
    HWND     hwndEntry;    /* Window handle                             */
```

```
    CHAR     buf[30];      /* Buffer                                    */

    switch( msg )
    {
        case WM_INITDLG:

            somSelf = (SOMAny *) mp2;   /* obtain creation parms */

            pwin = (PWINDATA) _wpAllocMem ( somSelf, sizeof (WINDATA), NULL );

            WinSetWindowPtr (hwndDlg, QWL_USER, pwin );

            /* Initialize the windata structure */
            pwin->somSelf     = somSelf;
            pwin->somClassObj = _somGetClass ( pwin->somSelf );

            /* query title string from instance data, and fill  */
            /* entryfield with the string                       */

            psz = _UserGetTitle ( pwin->somSelf );

            hwndEntry = WinWindowFromID ( hwndDlg, DLG_ENTRYFIELD );

            WinSetWindowText ( hwndEntry, psz );

            WinSetFocus      ( HWND_DESKTOP, hwndEntry );

            /* return TRUE to tell PM that we changed focus */
            return (MRESULT) TRUE ;

        case WM_DESTROY:

            pwin = (PWINDATA) WinQueryWindowPtr ( hwndDlg, QWL_USER );

            /* Save entryfield data to instance fields */
            hwndEntry = WinWindowFromID ( hwndDlg, DLG_ENTRYFIELD);

            WinQueryWindowText ( hwndEntry, sizeof (buf), (PCH) &buf );

            /* Save entryfield data to instance data */
            _UserSetTitle ( pwin->somSelf, (PSZ) &buf );

            /* Save instance data to OS2.INI file *now* */
            _wpSaveImmediate ( pwin -> somSelf );

            _wpFreeMem( pwin->somSelf,(PBYTE) pwin );
            return ( WinDefDlgProc( hwndDlg, msg, mp1, mp2 ) );
    }
    return ( WinDefDlgProc( hwndDlg, msg, mp1, mp2 ) );
}
```

wpSaveImmediate - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Example Code
Related Methods
Glossary

wpSaveLong

wpSaveLong - Syntax

This instance method is called to allow the object to save a 32-bit instance data value.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
PSZ           pszClass;  /*  Pointer to the class name. */
ULONG         ulKey;     /*  Class-defined identifier. */
ULONG         ulValue;   /*  Value (ULONG) to be stored. */
BOOL          rc;        /*  Success indicator. */

rc = _wpSaveLong(somSelf, pszClass, ulKey,
       ulValue);
```

wpSaveLong Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSaveLong Parameter - pszClass

pszClass (PSZ) - input
Pointer to the class name.

A pointer to a zero-terminated string that contains any unique string. The class name is recommended but not enforced.

wpSaveLong Parameter - ulKey

ulKey (ULONG) - input
Class-defined identifier.

A class-defined identifier that correlates to a particular instance data variable.

wpSaveLong Parameter - ulValue

ulValue (ULONG) - input
Value (ULONG) to be stored.

wpSaveLong Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSaveLong - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pszClass (PSZ) - input
Pointer to the class name.

A pointer to a zero-terminated string that contains any unique string. The class name is recommended but not enforced.

ulKey (ULONG) - input
Class-defined identifier.

A class-defined identifier that correlates to a particular instance data variable.

ulValue (ULONG) - input
Value (ULONG) to be stored.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSaveLong - Remarks

The saved 32-bit data value can be restored by issuing a call to wpRestoreLong.

wpSaveLong - Usage

This method can be called only during the processing of the wpSaveState method.

wpSaveLong - How to Override

This method is generally not overridden.

wpSaveLong - Related Methods

Related Methods

- wpRestoreData
- wpRestoreLong
- wpRestoreState
- wpRestoreString
- wpSaveData
- wpSaveDeferred
- wpSaveImmediate
- wpSaveLong
- wpSaveState
- wpSaveString

wpSaveLong - Example Code

This example demonstrates override of wpRestoreState to save and restore a ULONG.

```
SOM_Scope BOOL   SOMLINK wpRestoreState(LockBox *somSelf,
  ULONG ulReserved)
{
    ULONG  ulSecretCode;       /* Secret code */
    BOOL   fSuccess;           /* Success or failure flag */

    LockBoxData *somThis = LockBoxGetData(somSelf);
    LockBoxMethodDebug("LockBox","wpRestoreState");

    fSuccess = _wpRestoreLong ( somSelf       /* Pointer to thy self */
                      , "LockBox"             /* Class name */
                      , LOCKBOX_KEY_SECRET    /* Key of data to get */
                      , &ulSecretCode    );   /* Data returned here */

    if ( !fSuccess )
    {
        ulSecretCode = 0L;    /* On first restore, make secret code 0 */

        /* save initial code to OS2.INI file */

        _wpSaveLong ( somSelf                /* Pointer to me */
                    , "LockBox"              /* Class name */
                    , LOCKBOX_KEY_SECRET     /* Key of data */
                    , ulSecretCode );        /* Data */
    }
    return (parent_wpRestoreState(somSelf,ulReserved));
}
```

wpSaveLong - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpSaveState

wpSaveState - Syntax

This instance method is called to allow the object to save its state.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL          rc;       /*  Success indicator. */

rc = _wpSaveState(somSelf);
```

wpSaveState Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSaveState Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSaveState - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSaveState - Remarks

The saved state of the object is restored during the processing of the wpRestoreState method.

wpSaveState - Usage

This method is generally called by the system while it is processing either the wpClose or wpSaveImmediate methods. If an immediate save is required, the wpSaveImmediate method can be called, but it is not recommended. wpSaveDeferred should be used.

wpSaveState - How to Override

This method should be overridden by all classes that provide savable settings. An override of the wpRestoreState method is a prerequisite.

Override processing of this method typically includes a series of calls to any combination of the save state methods:

- wpSaveData
- wpSaveLong
- wpSaveString

wpSaveState - Related Methods

Related Methods

- wpRestoreData
- wpRestoreLong
- wpRestoreState
- wpRestoreString
- wpSaveData
- wpSaveDeferred
- wpSaveImmediate
- wpSaveLong
- wpSaveState
- wpSaveString

wpSaveState - Example Code

In this example, wpSaveState is overridden to save instance data from the OS2.INI file.

```
#define WPNBK_KEY        1

/*  Instance method to return title information */

SOM_Scope PSZ   SOMLINK UserGetTitle(nbk *somSelf)
```

```
{
    nbkData *somThis = nbkGetData(somSelf);
    nbkMethodDebug("nbk","UserGetTitle");

    return (PSZ) _title;
}

/* Override of wpSaveState to save our instance data in OS2.INI */

SOM_Scope BOOL   SOMLINK wpSaveState(nbk *somSelf)
{
    PSZ  psz;

    nbkData *somThis = nbkGetData(somSelf);
    nbkMethodDebug("nbk","wpSaveState");

    psz = _UserGetTitle ( somSelf );    /* Get data to be saved */

    /* Save our data to the OS2.INI file  */

     _wpSaveData ( somSelf              /* Pointer to us */
                 , "nbk"                /* Class name */
                 , WPNBK_KEY            /* Key for data */
                 , psz                  /* Pointer to data */
                 , 40 );                /* Length of data  */

    return (parent_wpSaveState(somSelf));  /* Let parent store data */
}
```

wpSaveState - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpSaveString

wpSaveString - Syntax

This instance method is called to allow the object to save an ASCIIZ instance data string.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
PSZ           pszClass;  /*  Pointer to the class name. */
ULONG         ulKey;     /*  Class-defined identifier. */
PSZ           pszValue;  /*  String to be stored. */
BOOL          rc;        /*  Success indicator. */
```

```
rc = _wpSaveString(somSelf, pszClass, ulKey,
       pszValue);
```

wpSaveString Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSaveString Parameter - pszClass

pszClass (PSZ) - input
Pointer to the class name.

A pointer to a zero-terminated string that contains any unique string. The class name is recommended but not enforced.

wpSaveString Parameter - ulKey

ulKey (ULONG) - input
Class-defined identifier.

A class-defined identifier that correlates to a particular instance data variable.

wpSaveString Parameter - pszValue

pszValue (PSZ) - input
String to be stored.

wpSaveString Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSaveString - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pszClass (PSZ) - input
Pointer to the class name.

A pointer to a zero-terminated string that contains any unique string. The class name is recommended but not enforced.

ulKey (ULONG) - input
Class-defined identifier.

A class-defined identifier that correlates to a particular instance data variable.

pszValue (PSZ) - input
String to be stored.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSaveString - Remarks

The saved ASCIIZ string can be restored by issuing a call to wpRestoreString.

wpSaveString - Usage

This method can be called only during the processing of the wpSaveState method.

wpSaveString - How to Override

This method is generally not overridden.

wpSaveString - Related Methods

Related Methods

- wpRestoreData
- wpRestoreLong
- wpRestoreState
- wpRestoreString
- wpSaveData
- wpSaveDeferred
- wpSaveImmediate

- wpSaveLong
- wpSaveState
- wpSaveString

wpSaveString - Example Code

This example demonstrates an override of wpRestoreState method to save string data.

```
SOM_Scope BOOL   SOMLINK wpRestoreState(nbk   *somSelf,
                                        ULONG ulReserved)
{
    BOOL fSuccess;       /* Success flag */
    ULONG ulCount;       /* Length of title string */
    UCHAR uchName[40];   /* Buffer for title */

    nbkData *somThis = nbkGetData(somSelf);
    nbkMethodDebug("nbk","wpRestoreState");

    ulCount = 40L;                              /* Size of input buffer */
    fSuccess = _wpRestoreString ( somSelf       /* Self pointer */
                      , (PSZ) "nbk"             /* Class name */
                      , WPNBK_KEY               /* Key for data */
                      , &uchName                /* String to receive data */
                      , &ulCount   );           /* Amount of data returned */

    if ( !fSuccess ) {    /* First time object is being restored */

        // initialize string to NULL
        strcpy ( uchName, "" );

        // save initial string to ini file
        _wpSaveString ( somSelf                 /* Self pointer */
                    , (PSZ) "nbk"               /* Class name */
                    , WPNBK_KEY                 /* Key of data */
                    , uchName     );            /* String to write */
    }

    _UserSetTitle ( somSelf, uchName );   /* Set an instance variable */

    return (parent_wpRestoreState(somSelf,ulReserved));
}
```

wpSaveString - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpScanSetupString

wpScanSetupString - Syntax

This instance method is called by wpSetup to parse the KEYWORDS and KEYWORD values from the setup string that is passed when the object is created.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;          /*  Pointer to the object on which the method is being invoked. */
PSZ           pszSetupString;   /*  Class-specific setup parameters for an object. */
PSZ           pszKey;           /*  Key to scan for. */
PSZ           pszValue;         /*  Buffer for the value. */
PULONG        pcbValue;         /*  Buffer length. */
BOOL          rc;               /*  Success indicator. */

rc = _wpScanSetupString(somSelf, pszSetupString,
       pszKey, pszValue, pcbValue);
```

wpScanSetupString Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpScanSetupString Parameter - pszSetupString

pszSetupString (PSZ) - input
Class-specific setup parameters for an object.

wpScanSetupString Parameter - pszKey

pszKey (PSZ) - input
Key to scan for.

wpScanSetupString Parameter - pszValue

pszValue (PSZ) - input
Buffer for the value.

If NULL, the size of the buffer required is returned in *pcbValue* .

wpScanSetupString Parameter - pcbValue

pcbValue (PULONG) - in/out
Buffer length.

input	The length of the buffer pointed to by *pszValue*. If *pszValue* is not NULL, this parameter must be initialized with the size of the buffer string (*pszValue*).
output	The actual length of the returned string, plus one for the NULL terminator, or the size of the buffer required if *pszValue* is NULL or the input value of *pcbValue* is too small.

wpScanSetupString Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion; key found.
FALSE	Error occurred; key not found or buffer too small.

wpScanSetupString - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pszSetupString (PSZ) - input
Class-specific setup parameters for an object.

pszKey (PSZ) - input
Key to scan for.

pszValue (PSZ) - input
Buffer for the value.

If NULL, the size of the buffer required is returned in *pcbValue*.

pcbValue (PULONG) - in/out
Buffer length.

input	The length of the buffer pointed to by *pszValue*. If *pszValue* is not NULL, this parameter must be initialized with the size of the buffer string (*pszValue*).
output	The actual length of the returned string, plus one for the NULL terminator, or the size of the buffer required if *pszValue* is NULL or the input value of *pcbValue* is too small.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion; key found.
FALSE	Error occurred; key not found or buffer too small.

wpScanSetupString - Remarks

If a semicolon is needed in the setup string, the escape character ^ can be used.

wpScanSetupString - Usage

This method is generally called from within an override of the wpSetup method.

wpScanSetupString - How to Override

This method is generally not overridden.

wpScanSetupString - Related Methods

Related Methods

- wpSetup
- wpclsNew

wpScanSetupString - Example Code

This example overrides _wpSetup, specifies setup strings, does some initialization, and writes some default data to the object's real filename. This will provide something to look at if the user drag/drops on an editor or selects the open/editor view.

```
SOM_Scope BOOL   SOMLINK myf_wpSetup(MYFILE *somSelf,
                 PSZ pszSetupString)
{
    MYFILEData *somThis = MYFILEGetData(somSelf);
    ULONG  cbBytesWritten;           /* Number of bytes written */
    APIRET rc;                       /* Return code */
    BOOL   fSuccess;                 /* Success flag */
    HFILE  hf;                       /* File handle */
    ULONG  ulAction;                 /* Action taken by DosOpen */
    CHAR   szObjectFilename[CCHMAXPATH];    /* Buffer for wpQueryRealName() */
    ULONG  cb  = sizeof(szObjectFilename);  /* Size of object */
    PSZ    pszDefaultText;           /* Default text */
    BOOL   rcParentCall;             /* Result of parent's method */
    CHAR   szValue[CCHMAXPATH+1];
    ULONG  cbBuffer;

    MYFILEMethodDebug("MYFILE","myf_wpSetup");

   /*   When the object is created from scratch, put some default
        text into the file on the hard disk                          */

   fSuccess =
   _wpQueryRealName(                /* query full-pathname of object's file */
      somSelf,                                 /* pointer to this object */
      szObjectFilename,                                /* return buffer */
      &cb,                                             /* sizeof buffer */
```

```
      TRUE);                           /* request fully qualified pathname? */

   if (fSuccess)
   {
      rc = DosOpen(szObjectFilename, &hf, &ulAction, 20L, FILE_NORMAL,
                   FILE_OPEN | FILE_CREATE,
                   OPEN_ACCESS_READWRITE | OPEN_SHARE_DENYNONE,
                   NULL);

      if (rc == NO_ERROR) {

         pszDefaultText = _clsQueryDefaultFileText(_MYFILE);
         DosWrite(hf, pszDefaultText, strlen(pszDefaultText), &cbBytesWritten);
         DosClose(hf);
      } /* endif */

/* New Setup Strings parsed and processed by this override:
 *
 *             SOUND=(YES/NO)    YES will invoke _soundAlarm method
 *                               NO  will do nothing */
   } else {
      _wpclsSetError( somSelf, MYFILE_QRN_FAILED );
   } /* endif (fSuccess) */

   rcParentCall = parent_wpSetup(somSelf,pszSetupString);

            /* Process setup strings we understand */

   cbBuffer = CCHMAXPATH;
   if ( _wpScanSetupString(somSelf, pszSetupString, "SOUND",
                           szValue,&cbBuffer) )
   {
      if ((szValue[0] == 'Y') && (szValue[1] == 'E') && (szValue[2] == 'S')) {
         _soundAlarm(somSelf);
      } /* endif */
   } /* endif */

  return( rcParentCall );
}
```

wpScanSetupString - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpSearchFolder

wpSearchFolder - Syntax

Searches a folder for the name specified.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder       *somSelf;       /*  Pointer to the object on which the method is being invoked. */
PSZ             pszName;       /*  Descriptive name of what is being searched. */
ULONG           ulSearchType;  /*  Type of search being conducted. */
ULONG           ulLen;         /*  The number of SEARCH_INFO structures passed in. */
PSEARCH_INFO    pInfo;         /*  Pointer to the SEARCH_INFO structures. */
WPFolder       *ResultFolder;  /*  The folder to contain the links to the matched items. */
BOOL            rc;            /*  Success indicator */

rc = _wpSearchFolder(somSelf, pszName, ulSearchType,
       ulLen, pInfo, ResultFolder);
```

wpSearchFolder Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpSearchFolder Parameter - pszName

pszName (PSZ) - input
Descriptive name of what is being searched.

If ulSearchType is SEARCH_ALL_FOLDERS then pszName must contain just a descriptive name without path information. Otherwise pszName must be a fully-qualified descriptive name starting with a drive letter (for example, C:\Folder1\Folder2).

wpSearchFolder Parameter - ulSearchType

ulSearchType (ULONG) - input
Type of search being conducted.

1	SEARCH_ALL_FOLDERS
2	SEARCH_THIS_FOLDER
3	SEARCH_THIS_TREE

wpSearchFolder Parameter - ulLen

ulLen (ULONG) - input
The number of SEARCH_INFO structures passed in.

wpSearchFolder Parameter - pInfo

pInfo (PSEARCH_INFO) - input
Pointer to the SEARCH_INFO structures.

Contains information on what classes of objects to search for.

wpSearchFolder Parameter - ResultFolder

ResultFolder (WPFolder *) - input
The folder to contain the links to the matched items.

wpSearchFolder Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Search was successful and something was found.
FALSE	Search was cancelled, nothing was found, or an error occurred.

wpSearchFolder - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

pszName (PSZ) - input
Descriptive name of what is being searched.

If ulSearchType is SEARCH_ALL_FOLDERS then pszName must contain just a descriptive name without path information. Otherwise pszName must be a fully-qualified descriptive name starting with a drive letter (for example, C:\Folder1\Folder2).

ulSearchType (ULONG) - input
Type of search being conducted.

1	SEARCH_ALL_FOLDERS
2	SEARCH_THIS_FOLDER
3	SEARCH_THIS_TREE

ulLen (ULONG) - input
The number of SEARCH_INFO structures passed in.

pInfo (PSEARCH_INFO) - input
Pointer to the SEARCH_INFO structures.

Contains information on what classes of objects to search for.

ResultFolder (WPFolder *) - input

The folder to contain the links to the matched items.

rc (BOOL) - returns
Success indicator

TRUE	Search was successful and something was found.
FALSE	Search was cancelled, nothing was found, or an error occurred.

wpSearchFolder - Remarks

If ulSearchType is SEARCH_ALL_FOLDERS then pszName is used to build up a new name of the form *:\pszName. Each valid drive letter is then substituted in for the *, and the search is performed as described below.

Beginning at the root directory, search for folder matches at the name level. For the example C:\Folder1\Folder2*, search the C: folder for all occurrences of Folder objects with the descriptive name Folder1. Then search those for all Folder objects of descriptive name Folder2.

Then within Folder2 search for all objects that are descended from the classes listed in the pInfo structures passed in and see if the descriptive name matches. If a match is found, consult the pInfo structure again to determine if the fSkipExtendedSearchCriteria flag is set. If the flag is not set, add the object to the result folder. If the flag is set, do an extended criteria match. If it matches, add the object to the results folder. If there are subfolders to be searched, recurse down into the subfolders to continue the search.

All matches are placed in the result folder as a link to the matched item.

wpSearchFolder - Usage

This method can be called at any time.

wpSearchFolder - How to Override

This method should not be overridden.

wpSearchFolder - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Glossary

wpSelectCell

wpSelectCell - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to select a given cell from a palette object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPalette     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
HWND           hwndPal;  /*  Palette window handle. */
PCELL          pCell;    /*  Cell pointer. */

wpSelectCell(somSelf, hwndPal, pCell);
```

wpSelectCell Parameter - somSelf

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

wpSelectCell Parameter - hwndPal

hwndPal (HWND) - input
Palette window handle.

wpSelectCell Parameter - pCell

pCell (PCELL) - input
Cell pointer.

wpSelectCell - Return Value

There is no return value for this method.

wpSelectCell - Parameters

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

hwndPal (HWND) - input
Palette window handle.

pCell (PCELL) - input
Cell pointer.

There is no return value for this method.

wpSelectCell - How to Override

This method can be overridden by all subclasses that wish to perform actions when a cell is selected.

wpSelectCell - Related Methods

Related Methods

- wpDragCell
- wpEditCell
- wpInitCellStructs
- wpPaintCell
- wpRedrawCell
- wpSetupCell

wpSelectCell - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetActionButtonStyle

wpSetActionButtonStyle - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets how the action buttons are displayed on the Toolbar.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG            ulStyle;  /*  Flag indicating the style of the action buttons. */
BOOL             rc;       /*  Success indicator. */

rc = _wpSetActionButtonStyle(somSelf, ulStyle);
```

wpSetActionButtonStyle Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpSetActionButtonStyle Parameter - ulStyle

ulStyle (ULONG) - input
Flag indicating the style of the action buttons.

ACTION_BUTTONS_MINI

ACTION_BUTTONS_NORMAL

ACTION_BUTTONS_OFF

ACTION_BUTTONS_TEXT

wpSetActionButtonStyle Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.

FALSE
Error occurred.

wpSetActionButtonStyle - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

ulStyle (ULONG) - input
Flag indicating the style of the action buttons.

ACTION_BUTTONS_MINI

ACTION_BUTTONS_NORMAL

ACTION_BUTTONS_OFF

ACTION_BUTTONS_TEXT

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetActionButtonStyle - How to Override

This method is generally not overridden.

wpSetActionButtonStyle - Related Methods

Related Methods

- wpRefreshDrawer
- wpSetActionButtonStyle

wpSetActionButtonStyle - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetAssociatedFileIcon

wpSetAssociatedFileIcon - Syntax

This method is specific to version 2.1, or higher, of the OS/2 operating system.

This instance method is called in order to give the data file a chance to set the icon that corresponds to its associated default view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDataFile     *somSelf;  /*  Pointer to the object on which the method is being invoked. */

wpSetAssociatedFileIcon(somSelf);
```

wpSetAssociatedFileIcon Parameter - somSelf

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

wpSetAssociatedFileIcon - Return Value

There is no return value for this method.

wpSetAssociatedFileIcon - Parameters

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

There is no return value for this method.

wpSetAssociatedFileIcon - Remarks

This method will not be called if the data file has a customized icon (that is, if its .ICON ea is set). This method is called when the association information about a data file has changed, such as its real name, its type, or its default view.

wpSetAssociatedFileIcon - How to Override

You should override this method if you do not want the shell to change the icon on your class of data file objects or if you want to set your own icon for your set of associations that fall outside of the range that the shell knows about. If you do override to set the icon, you should use _wpSetIcon.

Additional methods overriden in class: WPDataFile

- wpRestoreState
- wpSetDefaultView
- wpSetRealName
- wpSetType

These methods have been overriden to call wpSetAssociatedFileIcon on the data file if the object is initialized (except for wpRestoreState) and the icon on the data file is a default icon (OBJSTYLE_NOTDEFAULTICON is off).

Additional methods overriden in class: WPProgramFile

wpSetAssociatedFileIcon is overridden so that the program file icon that corresponds to the program and not to the associated view can be set.

wpSetAssociatedFileIcon - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Glossary

wpSetAssociationFilter

wpSetAssociationFilter - Syntax

This instance method is called to set an association of the program object to a data file object based on a file title filter.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgram     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
PSZ            pszFilter;  /*  Pointer to file title filters. */
BOOL           rc;         /*  Success indicator. */

rc = _wpSetAssociationFilter(somSelf, pszFilter);
```

wpSetAssociationFilter Parameter - somSelf

somSelf (WPProgram *) - input

Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

wpSetAssociationFilter Parameter - pszFilter

pszFilter (PSZ) - input
Pointer to file title filters.

A pointer to a string containing file title filters to associate. This string can contain several file title filters separated by a comma; for example:

```
pszFilter = "*.TXT, *.DOC"
```

wpSetAssociationFilter Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetAssociationFilter - Parameters

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

pszFilter (PSZ) - input
Pointer to file title filters.

A pointer to a string containing file title filters to associate. This string can contain several file title filters separated by a comma; for example:

```
pszFilter = "*.TXT, *.DOC"
```

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetAssociationFilter - Remarks

The association filter is used to designate this program as an available open view for data file objects that have a title that matches one of the association filters which are set. If a data file object matches a filter in a program object or program file object, the title of the program object or program file object appears in the data file object's Open cascade of its pop-up menu. A call to the wpSetAssociationFilter method causes the existing association filters for this object to be replaced. To determine the existing association filters that are set on this object, a call to the wpQueryAssociationFilter method can be made. The wpSetAssociationType method can be called to set an association based on the type of data file object.

wpSetAssociationFilter - Usage

This method can be called at any time in order to set an association of the program object to a data file object based on a file title filter.

wpSetAssociationFilter - How to Override

This method is generally not overridden.

wpSetAssociationFilter - Related Methods

Related Methods

- wpQueryAssociationType (WPProgram)
- wpQueryAssociationType (WPProgramFile)
- wpSetAssociationType (WPProgramFile)

wpSetAssociationFilter - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpSetAssociationFilter

wpSetAssociationFilter - Syntax

This instance method is called to set an association of the program object to a data file object based on a file title filter.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgramFile     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
PSZ                pszFilter;   /*  Pointer to file title filters. */
BOOL               rc;          /*  Success indicator. */

rc = _wpSetAssociationFilter(somSelf, pszFilter);
```

wpSetAssociationFilter Parameter - somSelf

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

wpSetAssociationFilter Parameter - pszFilter

pszFilter (PSZ) - input
Pointer to file title filters.

A pointer to a string containing file title filters to associate. This string can contain several file title filters separated by a comma; for example,

```
pszFilter = "*.TXT, *.DOC"
```

wpSetAssociationFilter Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetAssociationFilter - Parameters

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

pszFilter (PSZ) - input
Pointer to file title filters.

A pointer to a string containing file title filters to associate. This string can contain several file title filters separated by a comma; for example,

```
pszFilter = "*.TXT, *.DOC"
```

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetAssociationFilter - Remarks

The association filter is used to designate this program as an available open view for data file objects that have a title that matches one of the association filters which are set. If a data file object matches a filter in a program object or program file object, the title of the program object or program file object appears in the data file object's Open cascade of its pop-up menu. A call to the wpSetAssociationFilter method causes the existing association filters for this object to be replaced. To determine the existing association filters that are set on this object, a call to the wpQueryAssociationFilter method can be made. The wpSetAssociationType method can be called to set an association based on the type of data file object.

wpSetAssociationFilter - Usage

This method can be called at any time in order to set an association of the program object to a data file object based on a file title filter.

wpSetAssociationFilter - How to Override

This method is generally not overridden.

wpSetAssociationFilter - Related Methods

Related Methods

- wpQueryAssociationType (WPProgram)
- wpQueryAssociationType (WPProgramFile)
- wpSetAssociationType (WPProgram)

wpSetAssociationFilter - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpSetAssociationType

wpSetAssociationType - Syntax

This instance method is called to set an association of the program object to a data file object based on a file type.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgram     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PSZ            pszType;  /*  Pointer to file types. */
BOOL           rc;       /*  Success indicator. */

rc = _wpSetAssociationType(somSelf, pszType);
```

wpSetAssociationType Parameter - somSelf

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

wpSetAssociationType Parameter - pszType

pszType (PSZ) - input
Pointer to file types.

A pointer to a string containing file types to associate. This string can contain several file types separated by a comma; for example:

```
pszType = "in Text,C Code"
```

wpSetAssociationType Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.

FALSE	Error occurred.

wpSetAssociationType - Parameters

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

pszType (PSZ) - input
Pointer to file types.

A pointer to a string containing file types to associate. This string can contain several file types separated by a comma; for example:

```
pszType = "in Text,C Code"
```

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetAssociationType - Remarks

The association type is used to designate this program as an available open view for data file objects that have a type that matches one of the association types which are set. If a data file object matches a type in a program object or program file object, the title of the program object or program file object appears in the data file object's Open cascade of its pop-up menu. A call to the wpSetAssociationType method causes the existing association types for this object to be replaced. To determine the existing association types that are set on this object, a call to the wpQueryAssociationType method can be made. The wpSetAssociationFilter method can be called to set an association based on the title of the data file object.

wpSetAssociationType - Usage

This method can be called at any time in order to set an association of the program object to a data file object based on a file type.

wpSetAssociationType - How to Override

This method is generally not overridden.

wpSetAssociationType - Related Methods

Related Methods

- wpQueryAssociationType (WPProgram)
- wpQueryAssociationType (WPProgramFile)

- wpSetAssociationType (WPProgramFile

wpSetAssociationType - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpSetAssociationType

wpSetAssociationType - Syntax

This instance method is called to set an association of the program object to a data file object based on a file type.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgramFile     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PSZ                pszType;  /*  Pointer to file types. */
BOOL               rc;       /*  Success indicator. */

rc = _wpSetAssociationType(somSelf, pszType);
```

wpSetAssociationType Parameter - somSelf

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

wpSetAssociationType Parameter - pszType

pszType (PSZ) - input
Pointer to file types.

A pointer to a string containing file types to associate. This string can contain several file types separated by a comma; for example:

```
pszType = "Plain Text,C Code"
```

wpSetAssociationType Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetAssociationType - Parameters

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

pszType (PSZ) - input
Pointer to file types.

A pointer to a string containing file types to associate. This string can contain several file types separated by a comma; for example:

```
pszType = "Plain Text,C Code"
```

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetAssociationType - Remarks

The association type is used to designate this program as an available open view for data file objects that have a type that matches one of the association types which are set. If a data file object matches a type in a program object or program file object, the title of the program object or program file object appears in the data file object's Open cascade of its pop-up menu. A call to the wpSetAssociationType method causes the existing association types for this object to be replaced. To determine the existing association types that are set on this object, a call to the wpQueryAssociationType method can be made. The wpSetAssociationFilter method can be called to set an association based on the title of the data file object.

wpSetAssociationType - Usage

This method can be called at any time in order to set an association of the program object to a data file object based on a file type.

wpSetAssociationType - How to Override

This method is generally not overridden.

wpSetAssociationType - Related Methods

Related Methods

- wpQueryAssociationType (WPProgram)
- wpQueryAssociationType (WPProgramFile)
- wpSetAssociationType (WPProgram)

wpSetAssociationType - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpSetAttr

wpSetAttr - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the file attributes of a filesystem-based object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
ULONG             attrFile;  /*  File attributes. */
BOOL              rc;        /*  Success indicator. */

rc = _wpSetAttr(somSelf, attrFile);
```

wpSetAttr Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpSetAttr Parameter - attrFile

attrFile (ULONG) - input
File attributes.

wpSetAttr Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetAttr - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

attrFile (ULONG) - input
File attributes.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetAttr - How to Override

This method is generally not overridden.

wpSetAttr - Related Methods

Related Methods

- wpQueryAttr

wpSetAttr - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetAutoLockup

wpSetAutoLockup - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the value of the automatic lockup option.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
BOOL           bAutoLockup;  /*  The value of the automatic lock option. */

wpSetAutoLockup(somSelf, bAutoLockup);
```

wpSetAutoLockup Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpSetAutoLockup Return Value - bAutoLockup

bAutoLockup (BOOL) - returns
The value of the automatic lock option.

TRUE	Automatic lockup selected.
FALSE	Manual lockup selected.

wpSetAutoLockup - Return Value

There is no return value for this method.

wpSetAutoLockup - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

bAutoLockup (BOOL) - returns
The value of the automatic lock option.

TRUE	Automatic lockup selected.
FALSE	Manual lockup selected.

There is no return value for this method.

wpSetAutoLockup - Usage

This method can be called at any time in order to set the automatic lockup option for the current desktop.

wpSetAutoLockup - How to Override

This method is generally not overridden.

wpSetAutoLockup - Related Methods

Related Methods

- wpQueryAutoLockup
- wpQueryLockupAutoDim
- wpQueryLockupBackground
- wpQueryLockupFullScreen
- wpQueryLockupOnStartup
- wpQueryLockupTimeout

- wpSetLockupAutoDim
- wpSetLockupBackground
- wpSetLockupFullScreen
- wpSetLockupOnStartup
- wpSetLockupTimeout

wpSetAutoLockup - Example Code

This example lets you find the current Desktop object and sets the current automatic lockup option setting to TRUE.

```
WPDesktop    *Desktop;

Desktop = _wpclsQueryObjectFromPath("<WP_DESKTOP>");
_wpSetAutoLockup(Desktop,TRUE);
```

wpSetAutoLockup - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetAutoRefresh

wpSetAutoRefresh - Syntax

This method is specific to version 2.1, or higher, of the OS/2 operating system.

This instance method provides the ability to enable or disable the power object status window's automatic refresh option.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPower     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
BOOL         fAutoRefresh;  /*  Automatic refresh flag. */

wpSetAutoRefresh(somSelf, fAutoRefresh);
```

wpSetAutoRefresh Parameter - somSelf

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

wpSetAutoRefresh Parameter - fAutoRefresh

fAutoRefresh (BOOL) - input
Automatic refresh flag.

TRUE	Enable automatic refresh of power status window.
FALSE	Disable automatic refresh of power status window.

wpSetAutoRefresh - Return Value

There is no return value for this method.

wpSetAutoRefresh - Parameters

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

fAutoRefresh (BOOL) - input
Automatic refresh flag.

TRUE	Enable automatic refresh of power status window.
FALSE	Disable automatic refresh of power status window.

There is no return value for this method.

wpSetAutoRefresh - Related Methods

Related Methods

- wpAddPowerPage
- wpAddPowerViewPage
- wpChangePowerState
- wpQueryAutoRefresh
- wpQueryPowerConfirmation
- wpQueryPowerManagement
- wpQueryRefreshRate

- wpSetAutoRefresh
- wpSetPowerConfirmation
- wpSetPowerManagement
- wpSetRefreshRate

wpSetAutoRefresh - Topics

Select an item:
Syntax
Parameters
Returns
Related Methods
Glossary

wpSetBitmapData

wpSetBitmapData - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the bitmap data for the image file. This method will convert the bitmap data to the appropriate format and update the actual image data file.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPBitmap     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
PBYTE         pBitmapData;  /*  Buffer containing the image data in OS/2 Bitmap format. */
ULONG         ulSize;       /*  The size of the buffer. */
BOOL32        rc;           /*  Success indicator. */

rc = _wpSetBitmapData(somSelf, pBitmapData,
       ulSize);
```

wpSetBitmapData Parameter - somSelf

somSelf (WPBitmap *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPBitmap.

wpSetBitmapData Parameter - pBitmapData

pBitmapData (PBYTE) - input
Buffer containing the image data in OS/2 Bitmap format.

wpSetBitmapData Parameter - ulSize

ulSize (ULONG) - input
The size of the buffer.

wpSetBitmapData Return Value - rc

rc (BOOL32) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetBitmapData - Parameters

somSelf (WPBitmap *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPBitmap.

pBitmapData (PBYTE) - input
Buffer containing the image data in OS/2 Bitmap format.

ulSize (ULONG) - input
The size of the buffer.

rc (BOOL32) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetBitmapData - Remarks

This is a virtual method at the WPImageFile level. Subclasses that implement actual image file types are expected to subclass this method so they can return a bitmap representation of their data (for example, WPBitmap and MMImage).

wpSetBitmapData - Usage

The Workplace Shell calls the wpSetBitmapData method when it wants to change the contents of a WPImageFile object to contain a different image.

wpSetBitmapData - How to Override

The wpSetBitmapData method must be overridden in any subclass of WPImageFile.

wpSetBitmapData - Related Methods

Related Methods

- wpQueryBitmapData
- wpQueryBitmapHandle
- wpQueryBitmapInfoHeader
- wpReadImageFile
- wpWriteImageFile

wpSetBitmapData - Example Code

This example shows how the WPBitmap data class changes the contents of a bitmap file.

```
SOM_Scope BOOL32 bmp_wpSetBitmapData(WPImageFile *somSelf,
                                     PBYTE pBitmapData,
                                     ULONG ulSize)
{
    PBITMAPFILEHEADER      pbfh;
    BOOL                   bResult;

    /* Make sure this is really a bitmap definition  */
    pbfh = (PBITMAPFILEHEADER) pBitmapData;
    if (pbfh->usType != BFT_BITMAPARRAY &&
        pbfh->usType != BFT_BMAP)
    {
       return FALSE;
    }

    /* Free the old data buffer  */
    if(_pBitmapData)
    {
       _wpFreeMem(somSelf,_pBitmapData);
       _pBitmapData = NULL;
       _ulBitmapDataSize = 0;
    }

    /* Update the bitmap data file  */
    _pBitmapData = pBitmapData;
    _ulBitmapDataSize = ulSize;
    bResult = _wpWriteImageFile(somSelf);
    _pBitmapData = NULL;
    _ulBitmapDataSize = 0;

    /* If the file was written successfully, read the data back from the file
     * Note: This ensures that only the current resolution bitmap is
     *       kept in memory */
    if(bResult)
    {
       bResult = _wpReadImageFile(somSelf);
    }

    return bResult;
}
```

wpSetBitmapData - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetButtonAppearance

wpSetButtonAppearance - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the appearance of an object's Frame Control Button.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
ULONG         ulButtonType;  /*  Flag indicating the type of control button. */

wpSetButtonAppearance(somSelf, ulButtonType);
```

wpSetButtonAppearance Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetButtonAppearance Parameter - ulButtonType

ulButtonType (ULONG) - input
Flag indicating the type of control button.

Possible values are described in the following list:

HIDEBUTTON

MINBUTTON

DEFAULTBUTTON

wpSetButtonAppearance - Return Value

There is no return value for this method.

wpSetButtonAppearance - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulButtonType (ULONG) - input
Flag indicating the type of control button.

Possible values are described in the following list:

HIDEBUTTON

MINBUTTON

DEFAULTBUTTON

There is no return value for this method.

wpSetButtonAppearance - Remarks

This method is usually called by the system.

wpSetButtonAppearance - How to Override

This method is generally not overridden.

wpSetButtonAppearance - Related Methods

Related Methods

- wpQueryButtonAppearance
- wpclsQueryButtonAppearance

wpSetButtonAppearance - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpSetCloseDrawer

wpSetCloseDrawer - Syntax

This method is specific to Version 3, or higher, of the OS/2 operating system.

This instance method sets whether the drawers closes after an object in the drawers is opened from the drawers.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL             fState;   /*  Flag indicating whether to close the drawer after opening an object in the drawer

wpSetCloseDrawer(somSelf, fState);
```

wpSetCloseDrawer Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpSetCloseDrawer Parameter - fState

fState (BOOL) - input
Flag indicating whether to close the drawer after opening an object in the drawer.

TRUE
Close the drawer.
FALSE
Do not close the drawer.

wpSetCloseDrawer - Return Value

There is no return value for this method.

wpSetCloseDrawer - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

fState (BOOL) - input
Flag indicating whether to close the drawer after opening an object in the drawer.

TRUE
Close the drawer.
FALSE
Do not close the drawer.

There is no return value for this method.

wpSetCloseDrawer - Usage

This method can be called at any time to specify the behavior of the Toolbar drawers.

wpSetCloseDrawer - How to Override

This method is generally not overridden.

wpSetCloseDrawer - Related Methods

Related Methods

- wpQueryCloseDrawer
- wpRefreshDrawer

wpSetCloseDrawer - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpSetComputerName

wpSetComputerName - Syntax

This instance method is called to set the name of the computer that the printer exists on.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPrinter     *somSelf;          /*  Pointer to the object on which the method is being invoked. */
PSZ            pszComputerName;  /*  Computer name to be set. */
BOOL           rc;               /*  Success indicator. */

rc = _wpSetComputerName(somSelf, pszComputerName);
```

wpSetComputerName Parameter - somSelf

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

wpSetComputerName Parameter - pszComputerName

pszComputerName (PSZ) - input
Computer name to be set.

wpSetComputerName Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetComputerName - Parameters

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

pszComputerName (PSZ) - input
Computer name to be set.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetComputerName - How to Override

This method is generally not overridden.

wpSetComputerName - Related Methods

Related Methods

- wpQueryComputerName
- wpSetPrinterName

wpSetComputerName - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetConcurrentView

wpSetConcurrentView - Syntax

This instance method sets the multiple concurrent view behavior of the object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
ULONG         ulCCView;  /*  Flag indicating the concurrent view behavior of the object. */

wpSetConcurrentView(somSelf, ulCCView);
```

wpSetConcurrentView Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetConcurrentView Parameter - ulCCView

ulCCView (ULONG) - input
Flag indicating the concurrent view behavior of the object.

Possible values are described in the following list:

CCVIEW_DEFAULT
Concurrent view mode is set to the system default.

CCVIEW_ON
Concurrent view mode is on.

CCVIEW_OFF
Concurrent view mode is off.

wpSetConcurrentView - Return Value

There is no return value for this method.

wpSetConcurrentView - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulCCView (ULONG) - input
Flag indicating the concurrent view behavior of the object.

Possible values are described in the following list:

CCVIEW_DEFAULT
Concurrent view mode is set to the system default.
CCVIEW_ON
Concurrent view mode is on.
CCVIEW_OFF
Concurrent view mode is off.

There is no return value for this method.

wpSetConcurrentView - How to Override

This method is generally not overridden.

wpSetConcurrentView - Related Methods

Related Methods

- wpQueryConcurrentView

wpSetConcurrentView - Example Code

This example initializes the type of the instance of the Browse_O_Matic object to DLL.

```
SOM_Scope BOOL   SOMLINK Browse_O_Maticwps_wpSetup(Browse_O_Matic *somSelf,
                PSZ pszSetupString)
{
    /* Browse_O_MaticData *somThis = Browse_O_MaticGetData(somSelf); */
    Browse_O_MaticMethodDebug("Browse_O_Matic","Browse_O_Maticwps_wpSetup");

    _wpSetType( somSelf, DRT_DLL, NULL);

    /*********************************************/
    /* Allow mulitple secondary views of the object */
    /*********************************************/

    _wpSetConcurrentView( somSelf, CCVIEW_ON);

    return (parent_wpSetup(somSelf,pszSetupString));
}
```

wpSetConcurrentView - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Example Code
Related Methods
Glossary

wpSetContentsFromHandle

wpSetContentsFromHandle - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets a file's contents based on a handle to a known system data type. This is an abstract method; subclasses that describe files for which a system handle type exist (such as WPIcon or WPBitmap) should subclass this method.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDataFile     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
LHANDLE         handle;   /*  Handle to the data type object. */
BOOL            rc;       /*  Success indicator */

rc = _wpSetContentsFromHandle(somSelf, handle);
```

wpSetContentsFromHandle Parameter - somSelf

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

wpSetContentsFromHandle Parameter - handle

handle (LHANDLE) - input

Handle to the data type object.

wpSetContentsFromHandle Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetContentsFromHandle - Parameters

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

handle (LHANDLE) - input
Handle to the data type object.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetContentsFromHandle - Remarks

The handle returned by this method is data type specific. For example, an Image File object returns a handle to a bitmap (that is, an HBITMAP), while an Icon or Pointer might return a handle to a pointer (such as, an HPOINTER).

wpSetContentsFromHandle - Usage

This method can be called at any time to set the contents of a data file from a handle representing the appropriate type of object. For example, you can change the contents of any Image File object by calling the wpSetContentsFromHandle method with the handle of a bitmap.

wpSetContentsFromHandle - How to Override

This method should be overridden in any subclass of WPDataFile that can represent its data with a handle.

wpSetContentsFromHandle - Related Methods

Related Methods

- wpQueryHandleFromContents
- wpQueryPointerFromContents
- wpSetContentsFromPointer

wpSetContentsFromHandle - Example Code

This example shows how the Image File class processes the wpSetContentsFromHandle method. Notice how this method finds the bitmap data from the specified handle and calls the wpSetBitmapData method to update the data file. This allows any subclass of WPImageFile to provide this function without writing any extra code.

```
SOM_Scope BOOL SOMLINK img_wpSetContentsFromHandle(WPImageFile *somSelf,
                                                   LHANDLE handle)
{
   BITMAPINFOHEADER2    bmih;
   PBITMAPINFO2         pbmih = NULL;
   ULONG                scansize, tsize;
   LONG                 bmisize;
   PVOID                data;
   BOOL                 bResult;
   PBITMAPFILEHEADER2   pbmfh = NULL;
   ULONG                bmfhsize, ctabsize;
   HBITMAP              hbm = handle;
   HDC                  hdc = NULLHANDLE;
   SIZEL                sizl = {0L, 0L};
   HPS                  hps = NULLHANDLE;
   HPAL                 hPalette = NULLHANDLE;

   /* Get the bitmap information header from the bitmap  */
   bmih.cbFix = sizeof(bmih);
   if (!GpiQueryBitmapInfoHeader(hbm, &bmih))
   {
      return(FALSE);
   }

   /* Allocate a buffer large enough to contain the bitmap file header,
    * the bitmap information header, the color table, and the bitmap data */
   scansize = ((bmih.cBitCount*bmih.cx+31)/32)*bmih.cPlanes*4;
   bmisize = (bmih.cbFix+(sizeof(RGB2))*
                 (bmih.cclrUsed ? bmih.cclrUsed :
                                  1<<(bmih.cBitCount*bmih.cPlanes) ) );

   tsize = bmisize + scansize*bmih.cy + offsetof(BITMAPFILEHEADER2, bmp2);

   pbmfh = (PBITMAPFILEHEADER2) AllocMem(tsize, NULL);
   if (!pbmfh)
   {
      return(FALSE);
   }

   /* Copy the bitmap information header to the buffer  */
   pbmih = (PVOID) (((PBYTE)pbmfh) + offsetof(BITMAPFILEHEADER2,bmp2));
   data = (PVOID)pbmih;
   memcpy(data, (PVOID)&bmih, bmih.cbFix);
   pbmih->cBitCount = bmih.cBitCount*bmih.cPlanes;
   pbmih->cPlanes = 1;
   pbmih->cbImage = scansize*bmih.cy;

   /* Open a Device Context for the bitmap  */
   hdc = DevOpenDC (vhab, OD_MEMORY, "*", 0L, NULL, NULLHANDLE);
   if (hdc == DEV_ERROR)
   {
      FreeMem((PCHAR)pbmfh);
      pbmfh = NULL;
      return FALSE;
   }

   /* Create a Presentation Space for the bitmap  */
   hps = GpiCreatePS (vhab,
                      hdc,
                      &sizl,
                      PU_PELS | GPIT_NORMAL | GPIA_ASSOC);
```

```
   if (hps == GPI_ERROR)
   {
      DevCloseDC (hdc);
      hdc = NULLHANDLE;
      FreeMem((PCHAR)pbmfh);
      pbmfh = NULL;
      return FALSE;
   }

   /* Copy the color table and bitmap data to the buffer  */
   GpiCreateLogColorTable(hps,0,LCOLF_RGB,0,0,NULL);
   GpiSelectPalette(hps,hPalette);
   GpiSetBitmap(hps,hbm);
   GpiQueryBitmapBits(hps, 0, bmih.cy, ((PSZ)data)+bmisize, pbmih);
   GpiSetBitmap(hps,NULLHANDLE);
   GpiSelectPalette(hps,NULLHANDLE);
   GpiAssociate(hps,NULLHANDLE);
   GpiDestroyPS(hps);
   hps = NULLHANDLE;
   DevCloseDC (hdc);
   hdc = NULLHANDLE;

   ctabsize = ((sizeof(RGB2))* (pbmih->cclrUsed ? pbmih->cclrUsed :
                               1<<(pbmih->cBitCount*pbmih->cPlanes) ) );

   bmfhsize = offsetof(BITMAPFILEHEADER2, bmp2) +
              sizeof(BITMAPINFOHEADER2) +
              ctabsize;

   /* Construct the bitmap file header at the beginning of the buffer  */
   pbmfh->usType   = BFT_BMAP;
   pbmfh->cbSize   = bmfhsize+pbmfh->bmp2.cbImage;
   pbmfh->xHotspot = 0;
   pbmfh->yHotspot = 0;
   pbmfh->offBits  = bmfhsize;

   /* Update the image file from the bitmap data  */
   bResult = _wpSetBitmapData(somSelf,(PBYTE)pbmfh,tsize);

   /* Free the temporary buffer and return to the caller  */
   FreeMem((PCHAR)pbmfh);
   pbmfh = NULL;
   return bResult;
}
```

wpSetContentsFromHandle - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetContentsFromPointer

wpSetContentsFromPointer - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets a WPDataFile object's contents based on a pointer to shared memory.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDataFile     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PVOID           pData;    /*  Pointer to shared memory. */
BOOL            rc;       /*  Success indicator */

rc = _wpSetContentsFromPointer(somSelf, pData);
```

wpSetContentsFromPointer Parameter - somSelf

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

wpSetContentsFromPointer Parameter - pData

pData (PVOID) - input
Pointer to shared memory.

The first word must be a ULONG that contains the length, followed immediately by the data.

wpSetContentsFromPointer Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetContentsFromPointer - Parameters

somSelf (WPDataFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDataFile.

pData (PVOID) - input
Pointer to shared memory.

The first word must be a ULONG that contains the length, followed immediately by the data.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetContentsFromPointer - Usage

This method can be called at any time to change the contents of a data file.

wpSetContentsFromPointer - How to Override

This method is not generally overridden.

wpSetContentsFromPointer - Related Methods

Related Methods

- wpQueryHandleFromContents
- wpQueryPointerFromContents
- wpQueryPointerFromContents

wpSetContentsFromPointer - Example Code

This example copies the contents of one data file to another.

```
PVOID       pData;

pData = _wpQueryPointerFromContents(SourceDataFile);
if (pData)
{
   _wpSetContentsFromPointer(TargetDataFile,pData);
   DosFreeMem(pData);
}
```

wpSetContentsFromPointer - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code

wpSetCorrectDiskIcon

wpSetCorrectDiskIcon - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the correct icon for a given disk object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDisk     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL        rc;       /*  Success indicator. */

rc = _wpSetCorrectDiskIcon(somSelf);
```

wpSetCorrectDiskIcon Parameter - somSelf

somSelf (WPDisk *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDisk.

wpSetCorrectDiskIcon Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetCorrectDiskIcon - Parameters

somSelf (WPDisk *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDisk.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetCorrectDiskIcon - How to Override

This method can be overridden to set a different icon for the disk type.

wpSetCorrectDiskIcon - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Glossary

wpSetDateInfo

wpSetDateInfo - Syntax

This instance method is called to allow the object to change its current date information.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
FILEFINDBUF4     *pstFileFindBuf;  /*  Pointer to the file information buffer. */
ULONG             rc;              /*  Success indicator. */

rc = _wpSetDateInfo(somSelf, pstFileFindBuf);
```

wpSetDateInfo Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpSetDateInfo Parameter - pstFileFindBuf

pstFileFindBuf (FILEFINDBUF4 *) - input
Pointer to the file information buffer.

wpSetDateInfo Return Value - rc

rc (ULONG) - returns
Success indicator.

0
No error.
other
Error occurred.

wpSetDateInfo - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

pstFileFindBuf (FILEFINDBUF4 *) - input
Pointer to the file information buffer.

rc (ULONG) - returns
Success indicator.

0
No error.
other
Error occurred.

wpSetDateInfo - How to Override

This method is generally not overridden.

wpSetDateInfo - Topics

Select an item:
Syntax
Parameters
Returns
How to Override

wpSetDefaultDropOp

wpSetDefaultDropOp - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the default drop operation of objects that are dropped on the Desktop. The default is set on the Desktop page of the Desktop's Settings notebook.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;          /*  Pointer to the object on which the method is being invoked. */
ULONG          ulDefaultDropOp;
BOOL           rc;               /*  Success indicator */

rc = _wpSetDefaultDropOp(somSelf, ulDefaultDropOp);
```

wpSetDefaultDropOp Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpSetDefaultDropOp Parameter - ulDefaultDropOp

ulDefaultDropOp (ULONG) - input
Points to an object of class WPDesktop.

DO_LINK	Create shadow of dropped object
DO_MOVE	Move dropped object

wpSetDefaultDropOp Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetDefaultDropOp - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

ulDefaultDropOp (ULONG) - input
Points to an object of class WPDesktop.

DO_LINK	Create shadow of dropped object
DO_MOVE	Move dropped object

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetDefaultDropOp - Usage

This method can be called at any time in order to set the default drop operation for a desktop folder.

wpSetDefaultDropOp - How to Override

This method is generally not overridden.

wpSetDefaultDropOp - Related Methods

Related Methods

- wpQueryDefaultDropOp

wpSetDefaultDropOp - Example Code

This example lets you find the current Desktop folder and sets the default drop operation to DO_MOVE, which was the default for OS/2 Warp Version 3.

```
WPDesktop    *Desktop;

Desktop = _wpclsQueryObjectFromPath("<WP_DESKTOP>");
_wpSetDefaultDropOp(somSelf,DO_MOVE);
```

wpSetDefaultDropOp - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetDefaultHelp

wpSetDefaultHelp - Syntax

This instance method is called to set the default help panel for the object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
ULONG         HelpPanelID;  /*  Help panel identity. */
PSZ           HelpLibrary;  /*  Pointer to the name of the Help library. */
BOOL          rc;           /*  Success indicator. */

rc = _wpSetDefaultHelp(somSelf, HelpPanelID,
       HelpLibrary);
```

wpSetDefaultHelp Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetDefaultHelp Parameter - HelpPanelID

HelpPanelID (ULONG) - input
Help panel identity.

wpSetDefaultHelp Parameter - HelpLibrary

HelpLibrary (PSZ) - input
Pointer to the name of the Help library.

A NULL value implies that the default should be used.

wpSetDefaultHelp Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetDefaultHelp - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

HelpPanelID (ULONG) - input
Help panel identity.

HelpLibrary (PSZ) - input
Pointer to the name of the Help library.

A NULL value implies that the default should be used.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetDefaultHelp - Remarks

The default help panel for this class can be determined by calling the wpclsQueryDefaultHelp method.

wpSetDefaultHelp - Usage

This method can be called at any time in order to set the default help panel for this object.

wpSetDefaultHelp - How to Override

This method is generally not overridden.

wpSetDefaultHelp - Related Methods

Related Methods

- wpDisplayHelp
- wpMenuItemHelpSelected
- wpQueryDefaultHelp
- wpSetDefaultHelp
- wpclsQueryDefaultHelp

wpSetDefaultHelp - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpSetDefaultIconPos

wpSetDefaultIconPos - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to set the default icon position of an object within a folder.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PPOINTL       pPointl;  /*  Pointer to the coordinates of the default icon position. */
BOOL          rc;       /*  Success indicator. */

rc = _wpSetDefaultIconPos(somSelf, pPointl);
```

wpSetDefaultIconPos Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetDefaultIconPos Parameter - pPointl

pPointl (PPOINTL) - input
Pointer to the coordinates of the default icon position.

The *x* and *y* values represent the position in the object's folder in percentage coordinates.

wpSetDefaultIconPos Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetDefaultIconPos - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pPointl (PPOINTL) - input
Pointer to the coordinates of the default icon position.

The *x* and *y* values represent the position in the object's folder in percentage coordinates.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetDefaultIconPos - Remarks

Refer to wpSetup for more information about the initial icon position (ICONPOS).

wpSetDefaultIconPos - Usage

This method can be called at any time to set the default icon position.

wpSetDefaultIconPos - How to Override

This method is generally not overridden.

wpSetDefaultIconPos - Related Methods

Related Methods

- wpQueryDefaultIconPos

wpSetDefaultIconPos - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpSetDefaultPrinter

wpSetDefaultPrinter - Syntax

This instance method is called to set a default printer object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPrinter     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
```

```
BOOL            rc;       /*  Success indicator. */

rc = _wpSetDefaultPrinter(somSelf);
```

wpSetDefaultPrinter Parameter - somSelf

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

wpSetDefaultPrinter Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetDefaultPrinter - Parameters

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetDefaultPrinter - How to Override

This method is generally not overridden.

wpSetDefaultPrinter - Example Code

```
VOID Abort_All_Jobs( SOMAny *self )
```

```
{
   CHAR   szTitle[256];           /* Title of printer object */
   strcpy( szTitle, _wpQueryTitle( self ) );
   somPrintf("%s: Deleting all jobs.\n", szTitle);
   _wpSetDefaultPrinter( self );
   _wpHoldPrinter( self );
   _wpDeleteAllJobs( self );
}
```

wpSetDefaultPrinter - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Example Code
Glossary

wpSetDefaultView

wpSetDefaultView - Syntax

This instance method is called to allow an object to change its current default open view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         ulView;   /*  Specifies which view to open. */
BOOL          rc;       /*  Success indicator. */

rc = _wpSetDefaultView(somSelf, ulView);
```

wpSetDefaultView Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetDefaultView Parameter - ulView

ulView (ULONG) - input
Specifies which view to open.

OPEN_CONTENTS
Open contents view.
OPEN_DEFAULT
Open default view (same as double-click).
OPEN_DETAILS
Open details view.
OPEN_HELP
Display HelpPanel.
OPEN_RUNNING
Execute object.
OPEN_SETTINGS
Open Settings notebook.
OPEN_TREE
Open tree view.
OPEN_USER
Class-specific views have a greater value than this.

wpSetDefaultView Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetDefaultView - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulView (ULONG) - input
Specifies which view to open.

OPEN_CONTENTS
Open contents view.
OPEN_DEFAULT
Open default view (same as double-click).
OPEN_DETAILS
Open details view.
OPEN_HELP
Display HelpPanel.
OPEN_RUNNING
Execute object.
OPEN_SETTINGS
Open Settings notebook.
OPEN_TREE
Open tree view.

OPEN_USER
Class-specific views have a greater value than this.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetDefaultView - Remarks

The default open view for this class can be determined by calling the wpclsQueryDefaultView method.

wpSetDefaultView - Usage

This method can be called at any time in order to set the default open view for this object.

wpSetDefaultView - How to Override

This method is generally not overridden.

wpSetDefaultView - Related Methods

Related Methods

- wpQueryDefaultView
- wpclsQueryDefaultView
- wpSetDefaultView

wpSetDefaultView - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpSetDefStatusView

wpSetDefStatusView - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to set the default status view when Advanced Power Management (APM) is enabled.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPower     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
ULONG        ulDefStatus;  /*  Setting for the default status view. */

wpSetDefStatusView(somSelf, ulDefStatus);
```

wpSetDefStatusView Parameter - somSelf

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

wpSetDefStatusView Parameter - ulDefStatus

ulDefStatus (ULONG) - input
Setting for the default status view.

Possible values are described in the following list:

OPEN_STATUS
Full status of the object, including the battery life, power source, and battery state.

OPEN_BATTERY
Battery life indicator only.

wpSetDefStatusView - Return Value

There is no return value for this method.

wpSetDefStatusView - Parameters

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

ulDefStatus (ULONG) - input
Setting for the default status view.

Possible values are described in the following list:

OPEN_STATUS
Full status of the object, including the battery life, power source, and battery state.
OPEN_BATTERY
Battery life indicator only.

There is no return value for this method.

wpSetDefStatusView - Remarks

This method sets the default status view of the object while Advanced Power Management (APM) is enabled. If it is not enabled, this method sets the default status view for the next time APM is enabled.

wpSetDefStatusView - Usage

This method can be called at any time to set the default status view for this object.

wpSetDefStatusView - How to Override

This method is generally not overridden.

wpSetDefStatusView - Related Methods

Related Methods

- wpQueryDefStatusView
- wpQueryPowerManagement
- wpSetPowerManagement

wpSetDefStatusView - Topics

Select an item:
Syntax
Parameters
Returns
Remarks

How to Override
Usage
Related Methods
Glossary

wpSetDetailsColumnVisibility

wpSetDetailsColumnVisibility - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method either shows or hides the details column of a given folder.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         index;    /*  Column index for current details view. */
BOOL          Visible;  /*  Flag indicating whether to show or hide the details column. */
BOOL          rc;       /*  Success indicator. */

rc = _wpSetDetailsColumnVisibility(somSelf,
       index, Visible);
```

wpSetDetailsColumnVisibility Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpSetDetailsColumnVisibility Parameter - index

index (ULONG) - input
Column index for current details view.

wpSetDetailsColumnVisibility Parameter - Visible

Visible (BOOL) - input
Flag indicating whether to show or hide the details column.

Possible values are described in the following list:

TRUE
: Show the details column.

FALSE
: Hide the details column.

wpSetDetailsColumnVisibility Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
: Successful completion.

FALSE
: Error occurred.

wpSetDetailsColumnVisibility - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

index (ULONG) - input
Column index for current details view.

Visible (BOOL) - input
Flag indicating whether to show or hide the details column.

Possible values are described in the following list:

TRUE
: Show the details column.

FALSE
: Hide the details column.

rc (BOOL) - returns
Success indicator.

TRUE
: Successful completion.

FALSE
: Error occurred.

wpSetDetailsColumnVisibility - How to Override

This method is generally not overridden.

wpSetDetailsColumnVisibility - Related Methods

Related Methods

- wpIsDetailsColumnVisible

wpSetDetailsColumnVisibility - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetDisplaySmallIcons

wpSetDisplaySmallIcons - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets whether small or normal sized icons are displayed on the Toolbar and its drawers.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL             fState;   /*  Flag indicating whether to display small icons. */

wpSetDisplaySmallIcons(somSelf, fState);
```

wpSetDisplaySmallIcons Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpSetDisplaySmallIcons Parameter - fState

fState (BOOL) - input
Flag indicating whether to display small icons.

TRUE
Display small icons.
FALSE
Display normal-sized icons.

wpSetDisplaySmallIcons - Return Value

There is no return value for this method.

wpSetDisplaySmallIcons - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

fState (BOOL) - input
Flag indicating whether to display small icons.

TRUE
Display small icons.
FALSE
Display normal-sized icons.

There is no return value for this method.

wpSetDisplaySmallIcons - How to Override

This method is generally not overridden.

wpSetDisplaySmallIcons - Related Methods

Related Methods

- wpQueryDisplaySmallIcons
- wpRefreshDrawer

wpSetDisplaySmallIcons - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetDisplayText

wpSetDisplayText - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets whether the text is displayed under the icons on the Toolbar.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL             fState;   /*  Flag indicating whether to display text under icons. */

wpSetDisplayText(somSelf, fState);
```

wpSetDisplayText Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpSetDisplayText Parameter - fState

fState (BOOL) - input
Flag indicating whether to display text under icons.

TRUE
Display text.
FALSE
Do not display text.

wpSetDisplayText - Return Value

There is no return value for this method.

wpSetDisplayText - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

fState (BOOL) - input
Flag indicating whether to display text under icons.

TRUE
Display text.
FALSE
Do not display text.

There is no return value for this method.

wpSetDisplayText - How to Override

This method is generally not overridden.

wpSetDisplayText - Related Methods

Related Methods

- wpQueryDisplayText
- wpQueryDisplayTextInDrawers
- wpRefreshDrawer
- wpSetDisplayTextInDrawers

wpSetDisplayText - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetDisplayTextInDrawers

wpSetDisplayTextInDrawers - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets whether the text is displayed under the icons in Toolbar drawers.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL             fState;   /*  Flag indicating whether to display text under icons. */

wpSetDisplayTextInDrawers(somSelf, fState);
```

wpSetDisplayTextInDrawers Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpSetDisplayTextInDrawers Parameter - fState

fState (BOOL) - input
Flag indicating whether to display text under icons.

TRUE
Display text.

FALSE
Do not display text.

wpSetDisplayTextInDrawers - Return Value

There is no return value for this method.

wpSetDisplayTextInDrawers - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

fState (BOOL) - input
Flag indicating whether to display text under icons.

TRUE
Display text.
FALSE
Do not display text.

There is no return value for this method.

wpSetDisplayTextInDrawers - How to Override

This method is generally not overridden.

wpSetDisplayTextInDrawers - Related Methods

Related Methods

- wpQueryDisplayText
- wpQueryDisplayTextInDrawers
- wpRefreshDrawer
- wpSetDisplayText

wpSetDisplayTextInDrawers - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetDisplayVertical

wpSetDisplayVertical - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the orientation of the Toolbar.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL             fState;   /*  Flag indicating the orientation of the Toolbar. */

wpSetDisplayVertical(somSelf, fState);
```

wpSetDisplayVertical Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpSetDisplayVertical Parameter - fState

fState (BOOL) - input
Flag indicating the orientation of the Toolbar.

TRUE
Display the Toolbar vertically.
FALSE
Display the Toolbar horizontally.

wpSetDisplayVertical - Return Value

There is no return value for this method.

wpSetDisplayVertical - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

fState (BOOL) - input
Flag indicating the orientation of the Toolbar.

TRUE
Display the Toolbar vertically.
FALSE
Display the Toolbar horizontally.

There is no return value for this method.

wpSetDisplayVertical - Remarks

Drawers are displayed in the opposite direction.

wpSetDisplayVertical - How to Override

This method is generally not overridden.

wpSetDisplayVertical - Related Methods

Related Methods

- wpQueryDisplayVertical
- wpRefreshDrawer

wpSetDisplayVertical - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpSetDrawerHWND

wpSetDrawerHWND - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method stores the handle associated with the Toolbar or drawer into the instance data.

```
#define INCL_WINWORKPLACE
```

```
#include <os2.h>

WPLaunchPad     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
ULONG            ulDrawer;  /*  Toolbar or drawer being set. */
HWND             hwnd;      /*  Handle of the frame window for the drawer. */

wpSetDrawerHWND(somSelf, ulDrawer, hwnd);
```

wpSetDrawerHWND Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpSetDrawerHWND Parameter - ulDrawer

ulDrawer (ULONG) - input
Toolbar or drawer being set.

0
Toolbar.
>0
Drawer identifier.

wpSetDrawerHWND Parameter - hwnd

hwnd (HWND) - input
Handle of the frame window for the drawer.

A value of NULLHANDLE indicates that the view is closed.

wpSetDrawerHWND - Return Value

There is no return value for this method.

wpSetDrawerHWND - Parameters

somSelf (WPLaunchPad *) - input

Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

ulDrawer (ULONG) - input
Toolbar or drawer being set.

0
Toolbar.
>0
Drawer identifier.

hwnd (HWND) - input
Handle of the frame window for the drawer.

A value of NULLHANDLE indicates that the view is closed.

There is no return value for this method.

wpSetDrawerHWND - How to Override

This method can be overridden to know when a drawer view is opened.

wpSetDrawerHWND - Related Methods

Related Methods

- wpQueryDrawerHWND
- wpRefreshDrawer

wpSetDrawerHWND - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetError

wpSetError - Syntax

This instance method is called to identify an error condition.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
ULONG         ulErrorID;  /*  Error identity. */
BOOL          rc;         /*  Success indicator. */

rc = _wpSetError(somSelf, ulErrorID);
```

wpSetError Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetError Parameter - ulErrorID

ulErrorID (ULONG) - input
Error identity.

See the wpQueryError method for information about this value.

wpSetError Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetError - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulErrorID (ULONG) - input
Error identity.

See the wpQueryError method for information about this value.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetError - Remarks

wpSetError sets the last error on an object. The error identity is retrievable by issuing a call to the wpQueryError method.

wpSetError - Usage

This method can be called at any time in order to identify an error condition. This method is typically called prior to returning unsuccessfully from a method.

wpSetError - How to Override

This method is generally not overridden, unless the design of a customized error-handling system is planned.

wpSetError - Related Methods

Related Methods

- wpQueryError

wpSetError - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpSetFileSizeInfo

wpSetFileSizeInfo - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the file size values in the instance data.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
ULONG             cbFileSize;   /*  Size of the file. */
ULONG             cbEASize;     /*  Size of the file's extended attributes (EAs). */
ULONG             rc;           /*  Success indicator. */

rc = _wpSetFileSizeInfo(somSelf, cbFileSize,
       cbEASize);
```

wpSetFileSizeInfo Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpSetFileSizeInfo Parameter - cbFileSize

cbFileSize (ULONG) - input
Size of the file.

wpSetFileSizeInfo Parameter - cbEASize

cbEASize (ULONG) - input
Size of the file's extended attributes (EAs).

wpSetFileSizeInfo Return Value - rc

rc (ULONG) - returns
Success indicator.

0
No error.

other
Error occurred.

wpSetFileSizeInfo - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

cbFileSize (ULONG) - input
Size of the file.

cbEASize (ULONG) - input
Size of the file's extended attributes (EAs).

rc (ULONG) - returns
Success indicator.

0
No error.
other
Error occurred.

wpSetFileSizeInfo - How to Override

This method is generally not overridden.

wpSetFileSizeInfo - Related Methods

Related Methods

- wpQueryFileSize
- wpQueryEASize

wpSetFileSizeInfo - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetFldrAttr

wpSetFldrAttr - Syntax

This instance method is called to allow the folder to change its current view attributes.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         Attr;     /*  Flag indicating the object's attributes. */
ULONG         ulView;   /*  Flag indicating an object's open view. */
BOOL          rc;       /*  Success indicator. */

rc = _wpSetFldrAttr(somSelf, Attr, ulView);
```

wpSetFldrAttr Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpSetFldrAttr Parameter - Attr

Attr (ULONG) - input
Flag indicating the object's attributes.

These are the CV_* attributes defined by the Container Control Window. See CNRINFO for a detailed description.

wpSetFldrAttr Parameter - ulView

ulView (ULONG) - input
Flag indicating an object's open view.

OPEN_CONTENTS
Open contents view.
OPEN_DETAILS
Open details view.
OPEN_TREE
Open tree view.

wpSetFldrAttr Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetFldrAttr - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

Attr (ULONG) - input
Flag indicating the object's attributes.

These are the CV_* attributes defined by the Container Control Window. See CNRINFO for a detailed description.

ulView (ULONG) - input
Flag indicating an object's open view.

OPEN_CONTENTS
Open contents view.
OPEN_DETAILS
Open details view.
OPEN_TREE
Open tree view.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetFldrAttr - Remarks

The flags should be valid for the specified view. For example, CV_TREE should not be specified for the details view.

wpSetFldrAttr - Usage

This method can be called at any time in order to set the view attributes.

wpSetFldrAttr - How to Override

This method is generally not overridden.

wpSetFldrAttr - Related Methods

Related Methods

- wpQueryFldrAttr
- wpQueryFldrFlags
- wpQueryFldrFont
- wpSetFldrAttr
- wpSetFldrFlags
- wpSetFldrFont

wpSetFldrAttr - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpSetFldrBackground

wpSetFldrBackground - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the values for the folder background.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;           /*  Pointer to the object on which the method is being invoked. */
PSZ           pszImageFileName;  /*  The name of the background image file (or NULL if none is defined). */
ULONG         ulImageMode;       /*  The image mode. */
ULONG         ulScaleFactor;     /*  The scaling factor for scaled image mode. */
ULONG         ulBackgroundType;  /*  The background type. */
LONG          lBackgroundColor;  /*  The RGB value of the background color or USE_GLOBAL_COLOR. */

wpSetFldrBackground(somSelf, pszImageFileName,
       ulImageMode, ulScaleFactor, ulBackgroundType,
       lBackgroundColor);
```

wpSetFldrBackground Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpSetFldrBackground Parameter - pszImageFileName

pszImageFileName (PSZ) - output
The name of the background image file (or NULL if none is defined).

wpSetFldrBackground Parameter - ulImageMode

ulImageMode (ULONG) - output
The image mode.

BACKGROUND_NORMAL_IMAGE	Normal image.
BACKGROUND_TILED_IMAGE	Tiled image.
BACKGROUND_SCALED_IMAGE	Scaled image.

wpSetFldrBackground Parameter - ulScaleFactor

ulScaleFactor (ULONG) - output
The scaling factor for scaled image mode.

wpSetFldrBackground Parameter - ulBackgroundType

ulBackgroundType (ULONG) - output
The background type.

BACKGROUND_COLOR_ONLY	The background is color only (no image).
BACKGROUND_IMAGE	The background is an image.

wpSetFldrBackground Parameter - lBackgroundColor

lBackgroundColor (LONG) - output
The RGB value of the background color or USE_GLOBAL_COLOR.

The RGB value must be presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

wpSetFldrBackground - Return Value

There is no return value for this method.

wpSetFldrBackground - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

pszImageFileName (PSZ) - output
The name of the background image file (or NULL if none is defined).

ulImageMode (ULONG) - output
The image mode.

BACKGROUND_NORMAL_IMAGE	Normal image.
BACKGROUND_TILED_IMAGE	Tiled image.
BACKGROUND_SCALED_IMAGE	Scaled image.

ulScaleFactor (ULONG) - output
The scaling factor for scaled image mode.

ulBackgroundType (ULONG) - output
The background type.

BACKGROUND_COLOR_ONLY	The background is color only (no image).
BACKGROUND_IMAGE	The background is an image.

lBackgroundColor (LONG) - output
The RGB value of the background color or USE_GLOBAL_COLOR.

The RGB value must be presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

There is no return value for this method.

wpSetFldrBackground - Usage

This method can be called at any time in order to set the background for a folder.

wpSetFldrBackground - How to Override

This method is generally not overridden.

wpSetFldrBackground - Related Methods

Related Methods

- wpQueryFldrBackground

wpSetFldrBackground - Example Code

This exmaple sets the background for the current folder to WARPLOGO.BMP in the OS2\BITMAP directory on the boot drive as a tiled image.

```
_wpSetFldrBackground(somSelf,
                     "?:\\os2\\bitmap\\warplogo.bmp",
                     BACKGROUND_TILED_IMAGE,
                     1,
                     BACKGROUND_IMAGE,
                     CYAN);
```

wpSetFldrBackground - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetFldrDetailsClass

wpSetFldrDetailsClass - Syntax

This instance method is called to set the class for which details in the folder will be displayed.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder       *somSelf;  /*  Pointer to the object on which the method is being invoked. */
M_WPObject     *Class;    /*  Pointer to the class object for which details are to be displayed. */
BOOL            rc;       /*  Success indicator. */

rc = _wpSetFldrDetailsClass(somSelf, Class);
```

wpSetFldrDetailsClass Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpSetFldrDetailsClass Parameter - Class

Class (M_WPObject *) - input
Pointer to the class object for which details are to be displayed.

wpSetFldrDetailsClass Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetFldrDetailsClass - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

Class (M_WPObject *) - input
Pointer to the class object for which details are to be displayed.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetFldrDetailsClass - Remarks

Because folders can contain objects of different classes which can have different details, it is often necessary for the user to specify which class of details is to be displayed. The value set by this method is not used until a details view of the folder is opened. The wpQueryFldrDetailsClass method can be called to determine the class of details currently set. All column visibility states are reset by this method.

wpSetFldrDetailsClass - Usage

This method can be called at any time in order to set the current class of details to be displayed.

wpSetFldrDetailsClass - How to Override

This method is generally not overridden.

wpSetFldrDetailsClass - Related Methods

Related Methods

- wpQueryFldrDetailsClass

wpSetFldrDetailsClass - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpSetFldrFlags

wpSetFldrFlags - Syntax

This instance method is called to allow the folder to change its current flags.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         ulFlags;  /*  Folder flags to be set. */
BOOL          rc;       /*  Success indicator. */

rc = _wpSetFldrFlags(somSelf, ulFlags);
```

wpSetFldrFlags Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpSetFldrFlags Parameter - ulFlags

ulFlags (ULONG) - input
Folder flags to be set.

FOI_POPULATEDWITHALL
This flag is set if the folder was populated with all its contents.

FOI_POPULATEDWITHFOLDERS
This flag is set if the folder was populated only with folders that it contains.

FOI_WORKAREA
This flag is set if the user sets the work-area property.

wpSetFldrFlags Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetFldrFlags - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulFlags (ULONG) - input
Folder flags to be set.

FOI_POPULATEDWITHALL
This flag is set if the folder was populated with all its contents.

FOI_POPULATEDWITHFOLDERS
This flag is set if the folder was populated only with folders that it contains.

FOI_WORKAREA
This flag is set if the user sets the work-area property.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetFldrFlags - Usage

This method can be called at any time in order to set a folder's flag.

wpSetFldrFlags - How to Override

This method is generally not overridden.

wpSetFldrFlags - Related Methods

Related Methods

- wpQueryFldrAttr
- wpQueryFldrFlags
- wpQueryFldrFont
- wpSetFldrAttr
- wpSetFldrFlags
- wpSetFldrFont

wpSetFldrFlags - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpSetFldrFont

wpSetFldrFont - Syntax

This instance method is called to allow the folder to change its current font.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
```

```
PSZ           pszFont;  /*  Pointer to a buffer containing the font name. */
ULONG         ulView;   /*  Flag indicating an object's open view. */
BOOL          rc;       /*  Success indicator. */

rc = _wpSetFldrFont(somSelf, pszFont, ulView);
```

wpSetFldrFont Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpSetFldrFont Parameter - pszFont

pszFont (PSZ) - input
Pointer to a buffer containing the font name.

If pszFont is NULL, the font presentation paramete is set back to the default font. If a pointer to a NULL string is passed, the font presentation parameter will be set to the currently defined font.

The maximum font string size is set to PPFONTSTRSIZE.

wpSetFldrFont Parameter - ulView

ulView (ULONG) - input
Flag indicating an object's open view.

OPEN_CONTENTS
Open contents view.
OPEN_DETAILS
Open details view.
OPEN_TREE
Open tree view.

wpSetFldrFont Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetFldrFont - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

pszFont (PSZ) - input
Pointer to a buffer containing the font name.

If pszFont is NULL, the font presentation paramete is set back to the default font. If a pointer to a NULL string is passed, the font presentation parameter will be set to the currently defined font.

The maximum font string size is set to PPFONTSTRSIZE.

ulView (ULONG) - input
Flag indicating an object's open view.

OPEN_CONTENTS
Open contents view.
OPEN_DETAILS
Open details view.
OPEN_TREE
Open tree view.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetFldrFont - Remarks

The font name should be a valid presentation parameter string.

wpSetFldrFont - Usage

This method can be called at any time in order to change the current font for a view.

wpSetFldrFont - How to Override

This method is generally not overridden.

wpSetFldrFont - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Glossary

wpSetFldrSort

wpSetFldrSort - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the sort attributes on the folder window and saves those values in the instance data.

Note: This method only rearranges a folder open in icon view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
PVOID         pSortRecord;  /*  Pointer to the SORTFASTINFO structure containing sort record information. */
ULONG         ulView;       /*  Reserved - set to 0. */
ULONG         ulType;       /*  Reserved - set to 0. */
BOOL          rc;           /*  Success indicator. */

rc = _wpSetFldrSort(somSelf, pSortRecord,
       ulView, ulType);
```

wpSetFldrSort Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpSetFldrSort Parameter - pSortRecord

pSortRecord (PVOID) - input
Pointer to the SORTFASTINFO structure containing sort record information.

If this value is NULL, the sort attributes are reset to the default sort values.

wpSetFldrSort Parameter - ulView

ulView (ULONG) - input
Reserved - set to 0.

wpSetFldrSort Parameter - ulType

ulType (ULONG) - input
Reserved - set to 0.

wpSetFldrSort Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetFldrSort - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

pSortRecord (PVOID) - input
Pointer to the SORTFASTINFO structure containing sort record information.

If this value is NULL, the sort attributes are reset to the default sort values.

ulView (ULONG) - input
Reserved - set to 0.

ulType (ULONG) - input
Reserved - set to 0.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetFldrSort - Remarks

This method refreshes all open icon, tree, and details views of the folder with selected sort attributes.

wpSetFldrSort - How to Override

This method should not be overridden.

wpSetFldrSort - Related Methods

Related Methods

- wpQueryFldrSort
- wpQueryFldrSortClass
- wpSetFldrSortClass

wpSetFldrSort - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpSetFldrSortClass

wpSetFldrSortClass - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the sort class for the given folder.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder       *somSelf;  /*  Pointer to the object on which the method is being invoked. */
M_WPObject     *Class;    /*  Pointer to the sort class object. */
BOOL            rc;       /*  Success indicator. */

rc = _wpSetFldrSortClass(somSelf, Class);
```

wpSetFldrSortClass Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpSetFldrSortClass Parameter - Class

Class (M_WPObject *) - input
Pointer to the sort class object.

wpSetFldrSortClass Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetFldrSortClass - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

Class (M_WPObject *) - input
Pointer to the sort class object.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetFldrSortClass - Remarks

The sort class determines which class details can be used as candidates for the default or current sort field for a given folder (displayed in the sort conditional cascade submenu on the folder's context menu) as well as on the Sort Settings page of the object. The default sort class for all folders is WPFileSystem.

wpSetFldrSortClass - How to Override

This method is generally not overridden.

wpSetFldrSortClass - Related Methods

Related Methods

- wpQueryFldrSort
- wpQueryFldrSortClass
- wpSetFldrSort

wpSetFldrSortClass - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpSetFolder

wpSetFolder - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets this object to reside in the specified folder.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
WPObject     *container;  /*  Folder that the object is to reside in. */
BOOL          rc;         /*  Success indicator */

rc = _wpSetFolder(somSelf, container);
```

wpSetFolder Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetFolder Parameter - container

container (WPObject *) - input
Folder that the object is to reside in.

wpSetFolder Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetFolder - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

container (WPObject *) - input
Folder that the object is to reside in.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetFolder - Usage

This method is generally only called by the system.

wpSetFolder - How to Override

This method is generally not overridden.

wpSetFolder - Related Methods

Related Methods

- wpQueryFolder
- wpQueryOpenFolders

wpSetFolder - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpSetFloatOnTop

wpSetFloatOnTop - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets whether the Toolbar is forced to float on top of all other windows.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL             fState;   /*  Flag indicating whether to force the Toolbar to float on top of all windows. */

wpSetFloatOnTop(somSelf, fState);
```

wpSetFloatOnTop Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpSetFloatOnTop Parameter - fState

fState (BOOL) - input
Flag indicating whether to force the Toolbar to float on top of all windows.

TRUE
Force the Toolbar to float on top.
FALSE
Do not force the Toolbar to float on top.

wpSetFloatOnTop - Return Value

There is no return value for this method.

wpSetFloatOnTop - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

fState (BOOL) - input
Flag indicating whether to force the Toolbar to float on top of all windows.

TRUE
Force the Toolbar to float on top.
FALSE
Do not force the Toolbar to float on top.

There is no return value for this method.

wpSetFloatOnTop - How to Override

This method is generally not overridden.

wpSetFloatOnTop - Related Methods

Related Methods

- wpQueryFloatOnTop
- wpRefreshDrawer

wpSetFloatOnTop - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetHideLaunchPadFrameCtls

wpSetHideLaunchPadFrameCtls - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method hides the system menu and the title bar for the Toolbar.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL             fState;   /*  Flag indicating whether to hide frame controls. */

wpSetHideLaunchPadFrameCtls(somSelf, fState);
```

wpSetHideLaunchPadFrameCtls Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpSetHideLaunchPadFrameCtls Parameter - fState

fState (BOOL) - input
Flag indicating whether to hide frame controls.

TRUE
Hide the frame controls.

FALSE
Do not hide the frame controls.

wpSetHideLaunchPadFrameCtls - Return Value

There is no return value for this method.

wpSetHideLaunchPadFrameCtls - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

fState (BOOL) - input
Flag indicating whether to hide frame controls.

TRUE
Hide the frame controls.
FALSE
Do not hide the frame controls.

There is no return value for this method.

wpSetHideLaunchPadFrameCtls - How to Override

This method is generally not overridden.

wpSetHideLaunchPadFrameCtls - Related Methods

Related Methods

- wpQueryHideLaunchPadFrameCtls
- wpRefreshDrawer

wpSetHideLaunchPadFrameCtls - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetIcon

wpSetIcon - Syntax

This instance method is called to allow the object to set its current icon.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
HPOINTER      hptrNewIcon;  /*  Pointer to the object handle. */
BOOL          rc;           /*  Success indicator. */

rc = _wpSetIcon(somSelf, hptrNewIcon);
```

wpSetIcon Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetIcon Parameter - hptrNewIcon

hptrNewIcon (HPOINTER) - input
Pointer to the object handle.

wpSetIcon Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetIcon - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hptrNewIcon (HPOINTER) - input
Pointer to the object handle.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetIcon - Remarks

If the OBJSTYLE_NOTDEFAULTICON style is currently set for the object, the object's icon will be destroyed if the object is destroyed or made dormant.

wpSetIcon - Usage

This method can be called at any time in order to change the visible icon for this object. To permanently change the icon, the wpSetIconData method should be called.

wpSetIcon - How to Override

This method is generally not overridden.

wpSetIcon - Related Methods

Related Methods

- wpclsQueryIcon
- wpQueryIconData
- wpSetIcon
- wpSetIconData
- wpQueryIcon

wpSetIcon - Topics

Select an item:
Syntax
Parameters
Returns
Remarks

How to Override
Usage
Related Methods
Glossary

wpSetIconData

wpSetIconData - Syntax

This instance method is called to allow the object to permanently set its current icon.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
PICONINFO     pIconInfo;  /*  Pointer to an ICONINFO structure containing an icon specification. */
BOOL          rc;         /*  Success indicator. */

rc = _wpSetIconData(somSelf, pIconInfo);
```

wpSetIconData Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetIconData Parameter - pIconInfo

pIconInfo (PICONINFO) - input
Pointer to an ICONINFO structure containing an icon specification.

wpSetIconData Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetIconData - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pIconInfo (PICONINFO) - input
Pointer to an ICONINFO structure containing an icon specification.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetIconData - Usage

This method can be called at any time in order to permanently change the icon for this object. To temporarily change or refresh the icon for this object, the wpSetIcon method should be called.

wpSetIconData - How to Override

This method is generally not overridden.

wpSetIconData - Related Methods

Related Methods

- wpclsQueryIcon
- wpQueryIconData
- wpSetIcon
- wpSetIconData
- wpQueryIcon

wpSetIconData - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpSetIconTextBackgroundColor

wpSetIconTextBackgroundColor - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the color to be used for the icon text background.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;        /*  Pointer to the object on which the method is being invoked. */
ULONG         ulColor;        /*  The RGB color value or USE_GLOBAL_COLOR. */
BOOL          fRefreshViews;  /*  Refresh open views flag. */

wpSetIconTextBackgroundColor(somSelf, ulColor,
       fRefreshViews);
```

wpSetIconTextBackgroundColor Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpSetIconTextBackgroundColor Parameter - ulColor

ulColor (ULONG) - input
The RGB color value or USE_GLOBAL_COLOR.

The RGB value must be presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

wpSetIconTextBackgroundColor Parameter - fRefreshViews

fRefreshViews (BOOL) - input
Refresh open views flag.

TRUE	Refreshes all open views.
FALSE	Does not refresh the open views.

wpSetIconTextBackgroundColor - Return Value

There is no return value for this method.

wpSetIconTextBackgroundColor - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulColor (ULONG) - input
The RGB color value or USE_GLOBAL_COLOR.

The RGB value must be presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

fRefreshViews (BOOL) - input
Refresh open views flag.

TRUE	Refreshes all open views.
FALSE	Does not refresh the open views.

There is no return value for this method.

wpSetIconTextBackgroundColor - Usage

This method can be called at any time in order to set the text background color for the Icon, Tree, and Details View of a folder.

wpSetIconTextBackgroundColor - How to Override

This method is generally not overridden.

wpSetIconTextBackgroundColor - Related Methods

Related Methods

- wpQueryIconTextBackgroundColor
- wpQueryIconTextColor
- wpQueryIconTextVisibility
- wpQueryShadowTextColor
- wpSetIconTextColor
- wpSetIconTextVisibility
- wpSetShadowTextColor

wpSetIconTextBackgroundColor - Example Code

This example sets the text background color for the current folder to CYAN.

```
_wpSetIconTextBackgroundColor(somSelf,RGB_CYAN);
```

wpSetIconTextBackgroundColor - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetIconTextColor

wpSetIconTextColor - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the color to be used for the icon text for the specified view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;        /*  Pointer to the object on which the method is being invoked. */
ULONG         ulColor;        /*  The RGB color value or USE_GLOBAL_COLOR. */
ULONG         ulView;         /*  Specifies which view to query. */
BOOL          fRefreshViews;  /*  Refresh open views flag. */

wpSetIconTextColor(somSelf, ulColor, ulView,
       fRefreshViews);
```

wpSetIconTextColor Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpSetIconTextColor Parameter - ulColor

ulColor (ULONG) - input
The RGB color value or USE_GLOBAL_COLOR.

The RGB value must be presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

wpSetIconTextColor Parameter - ulView

ulView (ULONG) - input
Specifies which view to query.

OPEN_CONTENTS
Open contents view.
OPEN_DEFAULT
Open default view (same as double-click).
OPEN_DETAILS
Open details view.
OPEN_HELP
Display HelpPanel.
OPEN_RUNNING
Execute object.
OPEN_SETTINGS
Open Settings notebook.
OPEN_TREE
Open tree view.
OPEN_USER
Class-specific views have a greater value than this.

wpSetIconTextColor Parameter - fRefreshViews

fRefreshViews (BOOL) - input
Refresh open views flag.

TRUE	Refreshes all open views.
FALSE	Does not refresh the open views.

wpSetIconTextColor - Return Value

There is no return value for this method.

wpSetIconTextColor - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulColor (ULONG) - input
The RGB color value or USE_GLOBAL_COLOR.

The RGB value must be presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

ulView (ULONG) - input
Specifies which view to query.

OPEN_CONTENTS
Open contents view.
OPEN_DEFAULT
Open default view (same as double-click).
OPEN_DETAILS
Open details view.
OPEN_HELP
Display HelpPanel.
OPEN_RUNNING
Execute object.
OPEN_SETTINGS
Open Settings notebook.
OPEN_TREE
Open tree view.
OPEN_USER
Class-specific views have a greater value than this.

fRefreshViews (BOOL) - input
Refresh open views flag.

TRUE	Refreshes all open views.
FALSE	Does not refresh the open views.

There is no return value for this method.

wpSetIconTextColor - Usage

This method can be called at any time in order to set the text color for the titles displayed for normal icons in Icon, Tree, or Details View of a folder.

wpSetIconTextColor - How to Override

This method is generally not overridden.

wpSetIconTextColor - Related Methods

Related Methods

- wpQueryIconTextBackgroundColor
- wpQueryIconTextColor
- wpQueryIconTextVisibility
- wpQueryShadowTextColor
- wpSetIconTextBackgroundColor
- wpSetIconTextVisibility
- wpSetShadowTextColor

wpSetIconTextColor - Example Code

This example sets the icon text color for normal icons in Icon View for the current folder to BLACK.

```
_wpSetIconTextColor(somSelf,RGB_BLACK);
```

wpSetIconTextColor - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetIconTextVisibility

wpSetIconTextVisibility - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the icon text visibility for the specified view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;        /*  Pointer to the object on which the method is being invoked. */
ULONG         ulOption;       /*  The visibility state. */
ULONG         ulView;         /*  Specifies which view to query. */
BOOL          fRefreshViews;  /*  Refresh open views flag. */

wpSetIconTextVisibility(somSelf, ulOption,
```

```
ulView, fRefreshViews);
```

wpSetIconTextVisibility Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpSetIconTextVisibility Parameter - ulOption

ulOption (ULONG) - input
The visibility state.

0
Makes the icon text not visible.
ICON_TEXT_VISIBLE
Makes the icon text visible.
USE_GLOBAL_VISIBILITY
Uses the global value.

wpSetIconTextVisibility Parameter - ulView

ulView (ULONG) - input
Specifies which view to query.

OPEN_CONTENTS
Opens the Contents view.
OPEN_DEFAULT
Opens the default view (same as double-clicking).
OPEN_DETAILS
Opens the Details view.
OPEN_HELP
Displays the help panel.
OPEN_RUNNING
Executes the object.
OPEN_SETTINGS
Opens the Settings notebook.
OPEN_TREE
Opens the Tree view.
OPEN_USER
Class-specific views have a greater value than this.

wpSetIconTextVisibility Parameter - fRefreshViews

fRefreshViews (BOOL) - input

Refresh open views flag.

TRUE	Refresh all open views.
FALSE	Does not refresh the open views.

wpSetIconTextVisibility - Return Value

There is no return value for this method.

wpSetIconTextVisibility - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulOption (ULONG) - input
The visibility state.

0
Makes the icon text not visible.
ICON_TEXT_VISIBLE
Makes the icon text visible.
USE_GLOBAL_VISIBILITY
Uses the global value.

ulView (ULONG) - input
Specifies which view to query.

OPEN_CONTENTS
Opens the Contents view.
OPEN_DEFAULT
Opens the default view (same as double-clicking).
OPEN_DETAILS
Opens the Details view.
OPEN_HELP
Displays the help panel.
OPEN_RUNNING
Executes the object.
OPEN_SETTINGS
Opens the Settings notebook.
OPEN_TREE
Opens the Tree view.
OPEN_USER
Class-specific views have a greater value than this.

fRefreshViews (BOOL) - input
Refresh open views flag.

TRUE	Refresh all open views.
FALSE	Does not refresh the open views.

There is no return value for this method.

wpSetIconTextVisibility - Usage

This method can be called at any time in order to indicate whether or not the titles are displayed in Icon or Tree View of a folder.

wpSetIconTextVisibility - How to Override

This method is generally not overridden.

wpSetIconTextVisibility - Related Methods

Related Methods

- wpQueryIconTextBackgroundColor
- wpQueryIconTextColor
- wpQueryIconTextVisibility
- wpQueryShadowTextColor
- wpSetIconTextBackgroundColor
- wpSetIconTextColor
- wpSetShadowTextColor

wpSetIconTextVisibility - Example Code

This example sets the indiate that titles are to be displayed with the icons in Icon View of the current folder.

```
_wpSetIconTextVisibility(somSelf,ICON_TEXT_VISIBLE);
```

wpSetIconTextVisibility - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetJobProperties

wpSetJobProperties - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method is called to set up the default job properties of the printer object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPrinter     *self;       /*  Pointer to the object on which the method is being invoked. */
PDRIVDATA      pDrivData;  /*  Pointer to buffer containing PDRIVDATA (printer driver data) that this print obje
BOOL           rc;         /*  Success indicator */

rc = _wpSetJobProperties(self, pDrivData);
```

wpSetJobProperties Parameter - self

self (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

wpSetJobProperties Parameter - pDrivData

pDrivData (PDRIVDATA) - input
Pointer to buffer containing PDRIVDATA (printer driver data) that this print object is to use.

If this is NULL, this print object will display the print driver job properties dialog.

wpSetJobProperties Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetJobProperties - Parameters

self (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

pDrivData (PDRIVDATA) - input
Pointer to buffer containing PDRIVDATA (printer driver data) that this print object is to use.

If this is NULL, this print object will display the print driver job properties dialog.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetJobProperties - Remarks

This method displays or sets the job properties for this print pbject. If pDrivData is supplied (not NULL), then this method will set the job properties for the print object reference by itself. If pDrivData is not supplied (is NULL), then this method will display the job properties dialog to the user and then set the print object to have the new job properties.

wpSetJobProperties - Usage

This method can be called at any time in order to set the job properties of the printer object.

wpSetJobProperties - How to Override

This method is generally not overridden.

wpSetJobProperties - Related Methods

Related Methods

- wpDeleteAllJobs
- wpHoldPrinter

wpSetJobProperties - Example Code

This example creates a printer using default values and then sets the job properties of the newly created printer.

```
#include "wpprint.h"

WPPrinter *CreateAndSetPrintObject(WPPrinter *PrintObj)
{
  WPFolder *hwpFolder;

  hwpFolder = _wpclsQueryObjectFromPath(_WPFileSystem,
                                        (PSZ)"<WP_PRINTERSFOLDER>");
  if (!hwpFolder)
    return (PrintObj);

  if (!PrintObj)
```

```
    PrintObj = _wpclsNew(_WPPrinter,
                         "New Printer",
                         " ",
                         hwpFolder,
                         TRUE);

  if (PrintObj)
    _wpSetJobProperties(PrintObj,
                        NULL);

  return (PrintObj);
}
```

wpSetJobProperties - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetLinkToObject

wpSetLinkToObject - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method links the shadow object to the specified object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPShadow     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
WPObject     *FromObject;  /*  Pointer to the object to be linked to the shadow. */
BOOL          rc;          /*  Success indicator. */

rc = _wpSetLinkToObject(somSelf, FromObject);
```

wpSetLinkToObject Parameter - somSelf

somSelf (WPShadow *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPShadow.

wpSetLinkToObject Parameter - FromObject

FromObject (WPObject *) - input
Pointer to the object to be linked to the shadow.

wpSetLinkToObject Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetLinkToObject - Parameters

somSelf (WPShadow *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPShadow.

FromObject (WPObject *) - input
Pointer to the object to be linked to the shadow.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetLinkToObject - Usage

This method can be called at any time to link a shadow to an object.

wpSetLinkToObject - How to Override

This method is generally not overridden.

wpSetLinkToObject - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Glossary

wpSetLockupAutoDim

wpSetLockupAutoDim - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the value of the lockup auto-dim option.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
BOOL           bLockupAutoDim;  /*  The value of the lockup auto-dim options. */

wpSetLockupAutoDim(somSelf, bLockupAutoDim);
```

wpSetLockupAutoDim Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpSetLockupAutoDim Parameter - bLockupAutoDim

bLockupAutoDim (BOOL) - input
The value of the lockup auto-dim options.

TRUE	Clears the screen and displays the floating lock after the timeout.
FALSE	Does not clear the screen or display the floating lock after the timeout.

wpSetLockupAutoDim - Return Value

There is no return value for this method.

wpSetLockupAutoDim - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

bLockupAutoDim (BOOL) - input
The value of the lockup auto-dim options.

TRUE	Clears the screen and displays the floating lock after the timeout.
FALSE	Does not clear the screen or display the floating lock after the timeout.

There is no return value for this method.

wpSetLockupAutoDim - Usage

This method can be called at any time in order to set the lockup automatic dim option for the current desktop.

wpSetLockupAutoDim - How to Override

This method is generally not overridden.

wpSetLockupAutoDim - Related Methods

Related Methods

- wpQueryAutoLockup
- wpQueryLockupAutoDim
- wpQueryLockupBackground
- wpQueryLockupFullScreen
- wpQueryLockupOnStartup
- wpQueryLockupTimeout
- wpSetAutoLockup
- wpSetLockupBackground
- wpSetLockupFullScreen
- wpSetLockupOnStartup
- wpSetLockupTimeout

wpSetLockupAutoDim - Example Code

This example lets you find the current Desktop object and sets the current lockup automatic dim option setting to FALSE.

```
  WPDesktop    *Desktop;

  Desktop = _wpclsQueryObjectFromPath("<WP_DESKTOP>");
  _wpSetLockupAutoDim(Desktop,FALSE);
```

wpSetLockupAutoDim - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetLockupBackground

wpSetLockupBackground - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the current values for the lockup background.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;           /*  Pointer to the object on which the method is being invoked. */
PSZ            pszImageFileName;  /*  The name of the background image file (or NULL if none is defined). */
ULONG          ulImageMode;       /*  The following are the image modes: */
ULONG          ulScaleFactor;     /*  Scaling factor if the image mode is SCALED_IMAGE. */
ULONG          ulBackgroundType;  /*  Can be one of the following background types: */
LONG           lBackgroundColor;  /*  Background color as an encoded RGB value. */

wpSetLockupBackground(somSelf, pszImageFileName,
         ulImageMode, ulScaleFactor, ulBackgroundType,
         lBackgroundColor);
```

wpSetLockupBackground Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpSetLockupBackground Parameter - pszImageFileName

pszImageFileName (PSZ) - input
The name of the background image file (or NULL if none is defined).

wpSetLockupBackground Parameter - ulImageMode

ulImageMode (ULONG) - input
The following are the image modes:

BACKGROUND_NORMAL_IMAGE	Normal image.
BACKGROUND_TILED_IMAGE	Tiled image.
BACKGROUND_SCALED_IMAGE	Scaled image.

wpSetLockupBackground Parameter - ulScaleFactor

ulScaleFactor (ULONG) - input
Scaling factor if the image mode is SCALED_IMAGE.

wpSetLockupBackground Parameter - ulBackgroundType

ulBackgroundType (ULONG) - input
Can be one of the following background types:

BACKGROUND_COLOR_ONLY	Color only, no image.
BACKGROUND_IMAGE	Type of image (see image modes).

wpSetLockupBackground Parameter - lBackgroundColor

lBackgroundColor (LONG) - input
Background color as an encoded RGB value.

The RGB value must be presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

wpSetLockupBackground - Return Value

There is no return value for this method.

wpSetLockupBackground - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

pszImageFileName (PSZ) - input
The name of the background image file (or NULL if none is defined).

ulImageMode (ULONG) - input
The following are the image modes:

BACKGROUND_NORMAL_IMAGE	Normal image.
BACKGROUND_TILED_IMAGE	Tiled image.
BACKGROUND_SCALED_IMAGE	Scaled image.

ulScaleFactor (ULONG) - input
Scaling factor if the image mode is SCALED_IMAGE.

ulBackgroundType (ULONG) - input
Can be one of the following background types:

BACKGROUND_COLOR_ONLY	Color only, no image.
BACKGROUND_IMAGE	Type of image (see image modes).

lBackgroundColor (LONG) - input
Background color as an encoded RGB value.

The RGB value must be presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

There is no return value for this method.

wpSetLockupBackground - Usage

This method can be called at any time in order to set the lockup background for the current desktop.

wpSetLockupBackground - How to Override

This method is generally not overridden.

wpSetLockupBackground - Related Methods

Related Methods

- wpQueryAutoLockup
- wpQueryLockupAutoDim
- wpQueryLockupBackground
- wpQueryLockupFullScreen
- wpQueryLockupOnStartup
- wpQueryLockupTimeout
- wpSetAutoLockup
- wpSetLockupAutoDim
- wpSetLockupFullScreen
- wpSetLockupOnStartup
- wpSetLockupTimeout

wpSetLockupBackground - Example Code

This example lets you find the current Desktop object and sets the current lockup background to BACKGROUND_TILED_IMAGE and BACKGROUND_IMAGE.

```
WPDesktop    *Desktop;

Desktop = _wpclsQueryObjectFromPath("<WP_DESKTOP>");
_wpSetLockupBackground(Desktop,
                       "?:\\os2\\bitmap\\warplogo.bmp",
                       BACKGROUND_TILED_IMAGE,
                       1,
                       BACKGROUND_IMAGE,
                       CYAN);
```

wpSetLockupBackground - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetLockupFullScreen

wpSetLockupFullScreen - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the value of the lockup full-screen option.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;            /*  Pointer to the object on which the method is being invoked. */
BOOL           bLockupFullScreen;  /*  Value of the lockup full-screen option. */

wpSetLockupFullScreen(somSelf, bLockupFullScreen);
```

wpSetLockupFullScreen Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpSetLockupFullScreen Parameter - bLockupFullScreen

bLockupFullScreen (BOOL) - input
Value of the lockup full-screen option.

TRUE	Displays the full-screen lockup bitmap when locked.
FALSE	Displays the lockup window when locked.

wpSetLockupFullScreen - Return Value

There is no return value for this method.

wpSetLockupFullScreen - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

bLockupFullScreen (BOOL) - input
Value of the lockup full-screen option.

TRUE	Displays the full-screen lockup bitmap when locked.
FALSE	Displays the lockup window when locked.

There is no return value for this method.

wpSetLockupFullScreen - Usage

This method can be called at any time in order to set the lockup full-screen option for the current desktop.

wpSetLockupFullScreen - How to Override

This method is generally not overridden.

wpSetLockupFullScreen - Related Methods

Related Methods

- wpQueryLockupFullScreen

wpSetLockupFullScreen - Example Code

This example lets you find the current Desktop object and sets the current lockup full-screen option setting to TRUE.

```
WPDesktop    *Desktop;

Desktop = _wpclsQueryObjectFromPath("<WP_DESKTOP>");
_wpSetLockupFullScreen(Desktop,TRUE);
```

wpSetLockupFullScreen - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetLockupOnStartup

wpSetLockupOnStartup - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the value of the lockup on the startup option.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;           /*  Pointer to the object on which the method is being invoked. */
BOOL           bLockupOnStartup;  /*  Lockup value. */

wpSetLockupOnStartup(somSelf, bLockupOnStartup);
```

wpSetLockupOnStartup Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpSetLockupOnStartup Parameter - bLockupOnStartup

bLockupOnStartup (BOOL) - input
Lockup value.

TRUE	Automatically locks up the system when the system is started.
FALSE	The system is not locked up when started.

wpSetLockupOnStartup - Return Value

There is no return value for this method.

wpSetLockupOnStartup - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

bLockupOnStartup (BOOL) - input
Lockup value.

TRUE	Automatically locks up the system when the system is started.
FALSE	The system is not locked up when started.

There is no return value for this method.

wpSetLockupOnStartup - Usage

This method can be called at any time in order to set the lockup at the startup option for the current desktop.

wpSetLockupOnStartup - How to Override

This method is generally not overridden.

wpSetLockupOnStartup - Related Methods

Related Methods

- wpQueryAutoLockup
- wpQueryLockupAutoDim
- wpQueryLockupBackground
- wpQueryLockupFullScreen
- wpQueryLockupOnStartup
- wpQueryLockupTimeout
- wpSetAutoLockup
- wpSetLockupAutoDim
- wpSetLockupBackground
- wpSetLockupFullScreen
- wpSetLockupTimeout

wpSetLockupOnStartup - Example Code

This example lets you find the current Desktop object and sets the current lockup at startup option setting to FALSE.

```
WPDesktop    *Desktop;

Desktop = _wpclsQueryObjectFromPath("<WP_DESKTOP>");
_wpSetLockupOnStartup(Desktop,FALSE);
```

wpSetLockupOnStartup - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetLockupTimeout

wpSetLockupTimeout - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the timeout value for automatic lockup.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPDesktop     *somSelf;          /*  Pointer to the object on which the method is being invoked. */
ULONG          ulLockupTimeout;  /*  Automatic lockup timeout value in seconds. */

wpSetLockupTimeout(somSelf, ulLockupTimeout);
```

wpSetLockupTimeout Parameter - somSelf

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

wpSetLockupTimeout Parameter - ulLockupTimeout

ulLockupTimeout (ULONG) - input
Automatic lockup timeout value in seconds.

wpSetLockupTimeout - Return Value

There is no return value for this method.

wpSetLockupTimeout - Parameters

somSelf (WPDesktop *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPDesktop.

ulLockupTimeout (ULONG) - input
Automatic lockup timeout value in seconds.

There is no return value for this method.

wpSetLockupTimeout - Usage

This method can be called at any time in order to set the lockup automatic timeout value for the current desktop.

wpSetLockupTimeout - How to Override

This method is generally not overridden.

wpSetLockupTimeout - Related Methods

Related Methods

- wpQueryAutoLockup
- wpQueryLockupAutoDim
- wpQueryLockupBackground
- wpQueryLockupFullScreen
- wpQueryLockupOnStartup
- wpQueryLockupTimeout
- wpSetAutoLockup
- wpSetLockupAutoDim
- wpSetLockupBackground
- wpSetLockupFullScreen
- wpSetLockupOnStartup

wpSetLockupTimeout - Example Code

This example lets you find the current Desktop object and sets the current lockup automatic timeout value to 15 seconds.

```
WPDesktop    *Desktop;
```

```
Desktop = _wpclsQueryObjectFromPath("<WP_DESKTOP>");
_wpSetLockupTimeout(Desktop,15);
```

wpSetLockupTimeout - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetMenuBarVisibility

wpSetMenuBarVisibility - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the visibility state of the folder's menu bar.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
ULONG         ulVisibility;  /*  Visibility state. */
BOOL          rc;            /*  Success indicator */

rc = _wpSetMenuBarVisibility(somSelf, ulVisibility);
```

wpSetMenuBarVisibility Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpSetMenuBarVisibility Parameter - ulVisibility

ulVisibility (ULONG) - input
Visibility state.

MENUBAR_ON	Makes the menu bar visible.
MENUBAR_OFF	Makes the menu bar invisible.
MENUBAR_DEFAULT	Sets the menu bar visibility to the default state.

wpSetMenuBarVisibility Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetMenuBarVisibility - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulVisibility (ULONG) - input
Visibility state.

MENUBAR_ON	Makes the menu bar visible.
MENUBAR_OFF	Makes the menu bar invisible.
MENUBAR_DEFAULT	Sets the menu bar visibility to the default state.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetMenuBarVisibility - Usage

This method can be called at any time in order to indicate whether or not this folder displays menu bars.

wpSetMenuBarVisibility - How to Override

This method is generally not overridden.

wpSetMenuBarVisibility - Related Methods

Related Methods

- wpQueryMenuBarVisibility

wpSetMenuBarVisibility - Example Code

This example lets you indicate that the current folder is not to display a menu bar. This makes open views of the folder look similar to OS/2 Warp Version 3.

```
_wpSetMenuBarVisibility(somSelf,MENUBAR_OFF);
```

wpSetMenuBarVisibility - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetMenuStyle

wpSetMenuStyle - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the menu style to be either long or short.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         ulStyle;  /*  Menu style. */
BOOL          rc;       /*  Success indicator */

rc = _wpSetMenuStyle(somSelf, ulStyle);
```

wpSetMenuStyle Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetMenuStyle Parameter - ulStyle

ulStyle (ULONG) - input
Menu style.

MENUS_LONG	Long menus.
MENUS_SHORT	Short menus.
MENUS_DEFAULT	Sets the menu style to the default style.

wpSetMenuStyle Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetMenuStyle - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulStyle (ULONG) - input
Menu style.

MENUS_LONG	Long menus.
MENUS_SHORT	Short menus.
MENUS_DEFAULT	Sets the menu style to the default style.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetMenuStyle - Usage

This method can be called at any time in order to indicate whether this object displays long or short pop-up menus.

wpSetMenuStyle - How to Override

This method is generally not overridden.

wpSetMenuStyle - Related Methods

Related Methods

- wpQueryMenuStyle
- wpSetStyle

wpSetMenuStyle - Example Code

This example lets you indicate that the current object is to display short pop-up menus.

```
_wpSetMenuStyle(somSelf,MENUS_SHORT);
```

wpSetMenuStyle - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetMinWindow

wpSetMinWindow - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the minimized window behavior of an object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
ULONG         ulMinWindow;  /*  Flag indicating the minimized window behavior. */

wpSetMinWindow(somSelf, ulMinWindow);
```

wpSetMinWindow Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetMinWindow Parameter - ulMinWindow

ulMinWindow (ULONG) - input
Flag indicating the minimized window behavior.

Possible values are described in the following list:

MINWIN_DEFAULT
System default.
MINWIN_HIDDEN
Minimized windows are hidden.
MINWIN_VIEWER
Minimized windows are placed in the viewer.
MINWIN_DESKTOP
Minimized windows are placed on the desktop.

wpSetMinWindow - Return Value

There is no return value for this method.

wpSetMinWindow - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulMinWindow (ULONG) - input
Flag indicating the minimized window behavior.

Possible values are described in the following list:

MINWIN_DEFAULT
: System default.

MINWIN_HIDDEN
: Minimized windows are hidden.

MINWIN_VIEWER
: Minimized windows are placed in the viewer.

MINWIN_DESKTOP
: Minimized windows are placed on the desktop.

There is no return value for this method.

wpSetMinWindow - How to Override

This method is generally not overridden.

wpSetMinWindow - Related Methods

Related Methods

- wpQueryMinWindow

wpSetMinWindow - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetNetIdentity

wpSetNetIdentity - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the instance data containing the fully qualified network name of this network group.

```
#define INCL_WINWORKPLACE
#include <os2.h>
```

```
WPNetgrp     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
PSZ           pszNetIdentity;  /*  Fully qualified network name of this network group. */
BOOL          rc;              /*  Success indicator. */

rc = _wpSetNetIdentity(somSelf, pszNetIdentity);
```

wpSetNetIdentity Parameter - somSelf

somSelf (WPNetgrp *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPNetgrp.

wpSetNetIdentity Return Value - pszNetIdentity

pszNetIdentity (PSZ) - returns
Fully qualified network name of this network group.

For example, LSNetwork.

wpSetNetIdentity Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetNetIdentity - Parameters

somSelf (WPNetgrp *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPNetgrp.

pszNetIdentity (PSZ) - returns
Fully qualified network name of this network group.

For example, LSNetwork.

rc (BOOL) - returns
Success indicator.

TRUE

	Successful completion.
FALSE	Error occurred.

wpSetNetIdentity - How to Override

This method is generally not overridden.

wpSetNetIdentity - Related Methods

Related Methods

- wpQueryNetIdentity
- wpQueryObjectNetId
- wpQuerySrvrIdentity
- wpSetObjectNetId

wpSetNetIdentity - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetNextIconPos

wpSetNextIconPos - Syntax

This instance method is called to allow the folder to change the next icon position.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PPOINTL       pptl;     /*  Position within the folder. */
BOOL          rc;       /*  Success indicator. */

rc = _wpSetNextIconPos(somSelf, pptl);
```

wpSetNextIconPos Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpSetNextIconPos Parameter - pptl

pptl (PPOINTL) - input
Position within the folder.

wpSetNextIconPos Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetNextIconPos - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

pptl (PPOINTL) - input
Position within the folder.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetNextIconPos - Remarks

The next icon position is typically set during a drag or drop operation in order to ensure that the items dropped into a folder are located where the user wanted them.

wpSetNextIconPos - Usage

This method can be called at any time in order to set the next icon position at which objects will be inserted into the folder. If this method is used, the previous next position should be queried and restored afterwards.

wpSetNextIconPos - How to Override

This method is generally not overridden.

wpSetNextIconPos - Related Methods

Related Methods

- wpQueryNextIconPos

wpSetNextIconPos - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpSetObjectID

wpSetObjectID - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the object ID assigned to this object. The instance data is set, but the .INI file is not modified.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
PSZ           pszObjectID;  /*  String containing a unique object ID. */
BOOL          rc;           /*  Success indicator. */
```

```
rc = _wpSetObjectID(somSelf, pszObjectID);
```

wpSetObjectID Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetObjectID Parameter - pszObjectID

pszObjectID (PSZ) - input
String containing a unique object ID.

For example, "<PM_DESKTOP>".

wpSetObjectID Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetObjectID - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pszObjectID (PSZ) - input
String containing a unique object ID.

For example, "<PM_DESKTOP>".

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetObjectID - Remarks

The install routine uses the object ID to ensure that a given object is never re-installed, thereby avoiding duplication of objects.

The object ID is always lost when the object is copied; therefore, only one object ever exists within a given object ID.

wpSetObjectID - How to Override

This method is generally not overridden.

wpSetObjectID - Related Methods

Related Methods

- wpQueryObjectID

wpSetObjectID - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpSetObjectListFromHObjects

wpSetObjectListFromHObjects - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method adds a set of objects defined by a set of HOBJECTs to the Toolbar or drawer.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
ULONG            ulDrawer;      /*  Toolbar or drawer being set. */
```

```
ULONG            ulNumObjects;  /*  Number of objects pointed to by phobjects. */
HOBJECT         *phobjects;     /*  Pointer to a list of objects to be added. */
ULONG            ulAfter;       /*  Number of the object after which the new objects are inserted. */
BOOL             rc;            /*  Success indicator. */

rc = _wpSetObjectListFromHObjects(somSelf,
       ulDrawer, ulNumObjects, phobjects,
       ulAfter);
```

wpSetObjectListFromHObjects Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpSetObjectListFromHObjects Parameter - ulDrawer

ulDrawer (ULONG) - input
Toolbar or drawer being set.

0
Toolbar.
>0
Drawer identifier.

wpSetObjectListFromHObjects Parameter - ulNumObjects

ulNumObjects (ULONG) - input
Number of objects pointed to by *phobjects* .

wpSetObjectListFromHObjects Parameter - phobjects

phobjects (HOBJECT *) - in/out
Pointer to a list of objects to be added.

wpSetObjectListFromHObjects Parameter - ulAfter

ulAfter (ULONG) - input
Number of the object after which the new objects are inserted.

ADD_OBJECT_FIRST
: Adds the new objects to the top of the list.

ADD_OBJECT_LAST
: Adds the new objects to the bottom of the list.

other
: Adds the new objects after the object associated with the given index. For example, to insert after the first object, set *ulAfter* to 1.

wpSetObjectListFromHObjects Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
: Successful completion.

FALSE
: Error occurred.

wpSetObjectListFromHObjects - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

ulDrawer (ULONG) - input
Toolbar or drawer being set.

0
: Toolbar.

>0
: Drawer identifier.

ulNumObjects (ULONG) - input
Number of objects pointed to by *phobjects* .

phobjects (HOBJECT *) - in/out
Pointer to a list of objects to be added.

ulAfter (ULONG) - input
Number of the object after which the new objects are inserted.

ADD_OBJECT_FIRST
: Adds the new objects to the top of the list.

ADD_OBJECT_LAST
: Adds the new objects to the bottom of the list.

other
: Adds the new objects after the object associated with the given index. For example, to insert after the first object, set *ulAfter* to 1.

rc (BOOL) - returns
Success indicator.

TRUE
: Successful completion.

FALSE
: Error occurred.

wpSetObjectListFromHObjects - Remarks

Using HOBJECT for .INI files or files in which an application uses a rename/save/delete sequence is not supported.

wpSetObjectListFromHObjects - How to Override

This method can be overridden to know when objects are being added.

wpSetObjectListFromHObjects - Related Methods

Related Methods

- wpRefreshDrawer

wpSetObjectListFromHObjects - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpSetObjectListFromObjects

wpSetObjectListFromObjects - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method adds a set of objects defined by a set of objects to the Toolbar or drawer.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
ULONG            ulDrawer;      /*  Toolbar or drawer being set. */
```

```
ULONG            ulNumObjects;  /*  Number of objects pointed to by objects. */
WPObject       **objects;       /*  Pointer to an array of objects to be added. */
ULONG            ulAfter;       /*  Number of the object after which the new objects are inserted. */
BOOL             rc;            /*  Success indicator. */

rc = _wpSetObjectListFromObjects(somSelf,
       ulDrawer, ulNumObjects, objects, ulAfter);
```

wpSetObjectListFromObjects Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpSetObjectListFromObjects Parameter - ulDrawer

ulDrawer (ULONG) - input
Toolbar or drawer being set.

0
Toolbar.
>0
Drawer identifier.

wpSetObjectListFromObjects Parameter - ulNumObjects

ulNumObjects (ULONG) - input
Number of objects pointed to by *objects* .

wpSetObjectListFromObjects Parameter - objects

objects (WPObject **) - in/out
Pointer to an array of objects to be added.

wpSetObjectListFromObjects Parameter - ulAfter

ulAfter (ULONG) - input
Number of the object after which the new objects are inserted.

ADD_OBJECT_FIRST
Adds the new objects to the top of the list.

ADD_OBJECT_LAST
Adds the new objects to the bottom of the list.

other
Adds the new objects after the object associated with the given index. For example, to insert after the first object, set *ulAfter* to 1.

wpSetObjectListFromObjects Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetObjectListFromObjects - Parameters

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

ulDrawer (ULONG) - input
Toolbar or drawer being set.

0
Toolbar.
>0
Drawer identifier.

ulNumObjects (ULONG) - input
Number of objects pointed to by *objects* .

objects (WPObject **) - in/out
Pointer to an array of objects to be added.

ulAfter (ULONG) - input
Number of the object after which the new objects are inserted.

ADD_OBJECT_FIRST
Adds the new objects to the top of the list.

ADD_OBJECT_LAST
Adds the new objects to the bottom of the list.

other
Adds the new objects after the object associated with the given index. For example, to insert after the first object, set *ulAfter* to 1.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetObjectListFromObjects - How to Override

This method can be overridden to know when objects are being added.

wpSetObjectListFromObjects - Related Methods

Related Methods

- wpRefreshDrawer

wpSetObjectListFromObjects - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetObjectListFromStrings

wpSetObjectListFromStrings - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method adds a set of objects defined by a string list to the Toolbar or drawers.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPLaunchPad     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
ULONG            ulDrawer;  /*  Toolbar or drawers being set. */
PSZ              pszSetup;  /*  Setup string that defines the list of object IDs or the path and file names of ol
ULONG            ulAfter;   /*  Number of the object after which the new objects are inserted. */
BOOL             rc;        /*  Success indicator. */

rc = _wpSetObjectListFromStrings(somSelf,
       ulDrawer, pszSetup, ulAfter);
```

wpSetObjectListFromStrings Parameter - somSelf

somSelf (WPLaunchPad *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

wpSetObjectListFromStrings Parameter - ulDrawer

ulDrawer (ULONG) - input
Toolbar or drawers being set.

0
Toolbar.
>0
Drawer identifier.

wpSetObjectListFromStrings Parameter - pszSetup

pszSetup (PSZ) - input
Setup string that defines the list of object IDs or the path and file names of objects to be added.

Each entry in the list is terminated by a NULL character, and the last entry is terminted by two successive NULL characters.

wpSetObjectListFromStrings Parameter - ulAfter

ulAfter (ULONG) - input
Number of the object after which the new objects are inserted.

ADD_OBJECT_FIRST
Adds the new objects to the top of the list.

ADD_OBJECT_LAST
Adds the new objects to the bottom of the list.

other
Adds the new objects after the object associated with the given index. For example, to insert after the first object, set *ulAfter* to 1.

wpSetObjectListFromStrings Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.

FALSE
: Error occurred.

wpSetObjectListFromStrings - Parameters

somSelf (WPLaunchPad *) - input
: Pointer to the object on which the method is being invoked.

Points to an object of class WPLaunchPad.

ulDrawer (ULONG) - input
: Toolbar or drawers being set.

0
: Toolbar.

>0
: Drawer identifier.

pszSetup (PSZ) - input
: Setup string that defines the list of object IDs or the path and file names of objects to be added.

Each entry in the list is terminated by a NULL character, and the last entry is terminted by two successive NULL characters.

ulAfter (ULONG) - input
: Number of the object after which the new objects are inserted.

ADD_OBJECT_FIRST
: Adds the new objects to the top of the list.

ADD_OBJECT_LAST
: Adds the new objects to the bottom of the list.

other
: Adds the new objects after the object associated with the given index. For example, to insert after the first object, set *ulAfter* to 1.

rc (BOOL) - returns
: Success indicator.

TRUE
: Successful completion.

FALSE
: Error occurred.

wpSetObjectListFromStrings - How to Override

This method can be overridden to know when objects are being added.

wpSetObjectListFromStrings - Related Methods

Related Methods

- wpRefreshDrawer

wpSetObjectListFromStrings - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetObjectNetId

wpSetObjectNetId - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to set the network ID of the Shared Directory object that this object is linked to.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPNetLink     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
PSZ            pszNetIdentity;  /*  Pointer to the network ID string. */
BOOL           rc;              /*  Success indicator. */

rc = _wpSetObjectNetId(somSelf, pszNetIdentity);
```

wpSetObjectNetId Parameter - somSelf

somSelf (WPNetLink *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPNetLink.

wpSetObjectNetId Parameter - pszNetIdentity

pszNetIdentity (PSZ) - input
Pointer to the network ID string.

The network name is in the following format:

```
"<Network>\<Server>\<Share Name>"
```

Where:

<Network> Network identification
<Server> Server identification
<Shared Name>

wpSetObjectNetId Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetObjectNetId - Parameters

somSelf (WPNetLink *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPNetLink.

pszNetIdentity (PSZ) - input
Pointer to the network ID string.

The network name is in the following format:

```
"<Network>\<Server>\<Share Name>"
```

Where:

<Network> Network identification
<Server> Server identification
<Shared Name>

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetObjectNetId - Remarks

The Shared Directory object uses this ID to make its network calls. This ID is not stored in the object's data.

wpSetObjectNetId - Usage

This method can be called at any time to set the network ID.

wpSetObjectNetId - How to Override

This method is generally not overridden.

wpSetObjectNetId - Related Methods

Related Methods

- wpQueryNetIdentity
- wpQueryObjectNetId
- wpQuerySrvrIdentity

wpSetObjectNetId - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpSetPaletteInfo

wpSetPaletteInfo - Syntax

This instance method is called the set palette information.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPalette     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
PPALINFO       pPalInfo;  /*  Pointer to a PALINFO structure. */
BOOL           rc;        /*  Success indicator. */

rc = _wpSetPaletteInfo(somSelf, pPalInfo);
```

wpSetPaletteInfo Parameter - somSelf

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

wpSetPaletteInfo Parameter - pPalInfo

pPalInfo (PPALINFO) - input
Pointer to a PALINFO structure.

wpSetPaletteInfo Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetPaletteInfo - Parameters

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

pPalInfo (PPALINFO) - input
Pointer to a PALINFO structure.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetPaletteInfo - Remarks

The palette information can be retrieved by issuing a call to wpQueryPaletteInfo.

wpSetPaletteInfo - Usage

This method can be called at any time in order to set the palette information.

wpSetPaletteInfo - How to Override

This method is generally not overridden.

wpSetPaletteInfo - Related Methods

Related Methods

- wpQueryPaletteInfo

wpSetPaletteInfo - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpSetPowerConfirmation

wpSetPowerConfirmation - Syntax

This method is specific to version 2.1, or higher, of the OS/2 operating system.

This instance method provides the ability to enable or disable the power object confirmation message displayed when a Standby or Suspend request is made from the pop-up menu of the power object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPower     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
BOOL         fConfirm;  /*  Confirmation message. */

wpSetPowerConfirmation(somSelf, fConfirm);
```

wpSetPowerConfirmation Parameter - somSelf

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

wpSetPowerConfirmation Parameter - fConfirm

fConfirm (BOOL) - input
Confirmation message.

TRUE	Enable power confirmation message.
FALSE	Disable power confirmation message.

wpSetPowerConfirmation - Return Value

There is no return value for this method.

wpSetPowerConfirmation - Parameters

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

fConfirm (BOOL) - input
Confirmation message.

TRUE	Enable power confirmation message.
FALSE	Disable power confirmation message.

There is no return value for this method.

wpSetPowerConfirmation - Related Methods

Related Methods

- wpAddPowerPage
- wpAddPowerViewPage
- wpChangePowerState

- wpQueryAutoRefresh
- wpQueryPowerConfirmation
- wpQueryPowerManagement
- wpQueryRefreshRate
- wpSetAutoRefresh
- wpSetPowerConfirmation
- wpSetPowerManagement
- wpSetRefreshRate

wpSetPowerConfirmation - Topics

Select an item:
Syntax
Parameters
Returns
Related Methods
Glossary

wpSetPowerManagement

wpSetPowerManagement - Syntax

This method is specific to version 2.1, or higher, of the OS/2 operating system.

This instance method provides the ability to enable or disable the OS/2 Power Management support.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPower     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL         fPower;   /*  Power management flag. */
BOOL         rc;       /*  Success indicator. */

rc = _wpSetPowerManagement(somSelf, fPower);
```

wpSetPowerManagement Parameter - somSelf

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

wpSetPowerManagement Parameter - fPower

fPower (BOOL) - input
Power management flag.

TRUE	Enable power management.
FALSE	Disable power management.

wpSetPowerManagement Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetPowerManagement - Parameters

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

fPower (BOOL) - input
Power management flag.

TRUE	Enable power management.
FALSE	Disable power management.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetPowerManagement - Related Methods

Related Methods

- wpAddPowerPage
- wpAddPowerViewPage
- wpChangePowerState
- wpQueryAutoRefresh
- wpQueryPowerConfirmation
- wpQueryPowerManagement
- wpQueryRefreshRate
- wpSetAutoRefresh
- wpSetPowerConfirmation
- wpSetPowerManagement
- wpSetRefreshRate

wpSetPowerManagement - Topics

Select an item:
Syntax
Parameters
Returns
Related Methods
Glossary

wpSetPrinterName

wpSetPrinterName - Syntax

This instance method is called to set the name of the printer.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPrinter     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
PSZ            pszPrinterName;  /*  Printer name. */
BOOL           rc;              /*  Success indicator. */

rc = _wpSetPrinterName(somSelf, pszPrinterName);
```

wpSetPrinterName Parameter - somSelf

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

wpSetPrinterName Parameter - pszPrinterName

pszPrinterName (PSZ) - input
Printer name.

wpSetPrinterName Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetPrinterName - Parameters

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

pszPrinterName (PSZ) - input
Printer name.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetPrinterName - How to Override

This method is generally not overridden.

wpSetPrinterName - Related Methods

Related Methods

- wpQueryPrinterName
- wpSetComputerName

wpSetPrinterName - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetProgDetails

wpSetProgDetails - Syntax

This instance method is called to allow the object to change its program details.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgram      *somSelf;       /*  Pointer to the object on which the method is being invoked. */
PPROGDETAILS    pProgDetails;  /*  Pointer to the program details. */
BOOL            rc;            /*  Success indicator. */

rc = _wpSetProgDetails(somSelf, pProgDetails);
```

wpSetProgDetails Parameter - somSelf

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

wpSetProgDetails Parameter - pProgDetails

pProgDetails (PPROGDETAILS) - input
Pointer to the program details.

wpSetProgDetails Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetProgDetails - Parameters

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

pProgDetails (PPROGDETAILS) - input

Pointer to the program details.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetProgDetails - Usage

This method can be called at any time in order to set the details for this object.

wpSetProgDetails - How to Override

This method is generally not overridden.

wpSetProgDetails - Related Methods

Related Methods

- wpQueryProgDetails

wpSetProgDetails - Example Code

```
PPROGDETAILS pProgDetails;
ULONG        ulSize;

    /* Change the executable of a program object to be VIEW.EXE */

    if ((_wpQueryProgDetails (self, pProgDetails, &ulSize)))   {

      pProgDetails->pszExecutable = "VIEW.EXE";

      if ((rc = _wpSetProgDetails (self, pProgDetails))) {

        somPrintf("Program changed to VIEW.EXE\n");

      } else {

        somPrintf("_wpSetProgDetails failed.\n");
        return 1;

      } /* endif */

    } else {

      somPrintf("_wpQueryProgDetails failed.\n");
      return 1;

    } /* endif */
```

wpSetProgDetails - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetProgDetails

wpSetProgDetails - Syntax

This instance method is called to allow the object to change its program details.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgramFile     *somSelf;       /*  Pointer to the object on which the method is being invoked. */
PPROGDETAILS       pProgDetails;  /*  Pointer to the program details. */
BOOL               rc;            /*  Success indicator. */

rc = _wpSetProgDetails(somSelf, pProgDetails);
```

wpSetProgDetails Parameter - somSelf

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

wpSetProgDetails Parameter - pProgDetails

pProgDetails (PPROGDETAILS) - input
Pointer to the program details.

wpSetProgDetails Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetProgDetails - Parameters

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

pProgDetails (PPROGDETAILS) - input
Pointer to the program details.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetProgDetails - Usage

This method can be called at any time in order to set the details for this object.

wpSetProgDetails - How to Override

This method is generally not overridden.

wpSetProgDetails - Related Methods

Related Methods

- wpQueryProgDetails

wpSetProgDetails - Example Code

```
PPROGDETAILS pProgDetails;
ULONG        ulSize;

    /* Change the executable of a program object to be VIEW.EXE */


    if ((_wpQueryProgDetails (self, pProgDetails, &ulSize)))   {

      pProgDetails->pszExecutable = "VIEW.EXE";

      if ((rc = _wpSetProgDetails (self, pProgDetails))) {

        somPrintf("Program changed to VIEW.EXE\n");

      } else {

        somPrintf("_wpSetProgDetails failed.\n");
        return 1;

      } /* endif */

    } else {

      somPrintf("_wpQueryProgDetails failed.\n");
      return 1;

    } /* endif */
```

wpSetProgDetails - Topics

Select an item:

Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetProgIcon

wpSetProgIcon - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the visual icon for the current program to the appropriate default or custom icon.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgram     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PFEA2LIST      pfeal;    /*  Pointer to a list of FEA2 structures containing icon extended attributes (EAs). */
BOOL           rc;       /*  Success indicator. */

rc = _wpSetProgIcon(somSelf, pfeal);
```

wpSetProgIcon Parameter - somSelf

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

wpSetProgIcon Parameter - pfeal

pfeal (PFEA2LIST) - input
Pointer to a list of FEA2 structures containing icon extended attributes (EAs).

A value of NULL indicates that icon EAs are not available.

wpSetProgIcon Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetProgIcon - Parameters

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

pfeal (PFEA2LIST) - input
Pointer to a list of FEA2 structures containing icon extended attributes (EAs).

A value of NULL indicates that icon EAs are not available.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetProgIcon - Usage

This method can be called at any time.

wpSetProgIcon - How to Override

This method is generally not overridden.

wpSetProgIcon - Related Methods

Related Methods

- wpRestoreState
- wpSetProgDetails
- wpSetProgIcon (WPProgramFile)

wpSetProgIcon - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpSetProgIcon

wpSetProgIcon - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the visual icon for this program to the appropriate custom or default icon.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgramFile     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PFEA2LIST          pfeal;    /*  Pointer to a list. */
BOOL               rc;       /*  Success indicator. */
```

```
rc = _wpSetProgIcon(somSelf, pfeal);
```

wpSetProgIcon Parameter - somSelf

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

wpSetProgIcon Parameter - pfeal

pfeal (PFEA2LIST) - input
Pointer to a list.

Points to a list of FEA2 structures containing icon extended attributes (EAs). A value of NULL indicates that icon EAs are not available.

wpSetProgIcon Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetProgIcon - Parameters

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

pfeal (PFEA2LIST) - input
Pointer to a list.

Points to a list of FEA2 structures containing icon extended attributes (EAs). A value of NULL indicates that icon EAs are not available.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetProgIcon - Usage

This method can be called at any time to set an icon for a given program file.

wpSetProgIcon - How to Override

This method is generally not overridden.

wpSetProgIcon - Related Methods

Related Methods

- wpRestoreState
- wpSetProgDetails
- wpSetProgIcon (WPProgram)

wpSetProgIcon - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpSetProgramAssociations

wpSetProgramAssociations - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the program associations in the .INI file.

```
#define INCL_WINWORKPLACE
#include <os2.h>
```

```
WPProgram     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
PSZ            pszAssoc;  /*  Pointer to string of datafile association types or filters, separated by commas. *
BOOL           fFilter;   /*  Filter flag. */
BOOL           rc;        /*  Success indicator. */

rc = _wpSetProgramAssociations(somSelf, pszAssoc,
       fFilter);
```

wpSetProgramAssociations Parameter - somSelf

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

wpSetProgramAssociations Parameter - pszAssoc

pszAssoc (PSZ) - input
Pointer to string of datafile association types or filters, separated by commas.

wpSetProgramAssociations Parameter - fFilter

fFilter (BOOL) - input
Filter flag.

TRUE	*pszAssoc* contains filters.
FALSE	*pszAssoc* contains types.

wpSetProgramAssociations Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetProgramAssociations - Parameters

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

pszAssoc (PSZ) - input
Pointer to string of datafile association types or filters, separated by commas.

fFilter (BOOL) - input
Filter flag.

TRUE	*pszAssoc* contains filters.
FALSE	*pszAssoc* contains types.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetProgramAssociations - Usage

This method can be called at any time.

wpSetProgramAssociations - How to Override

This method is generally not overridden.

wpSetProgramAssociations - Related Methods

Related Methods

- wpQueryProgramAssociations

wpSetProgramAssociations - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpSetProgramAssociations

wpSetProgramAssociations - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method returns the program associations from the .INI file.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgramFile     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
PSZ                pszAssoc;  /*  Pointer to string of datafile association types or filters, separated by comma
BOOL               fFilter;   /*  Filter flag. */
BOOL               rc;        /*  Success indicator. */

rc = _wpSetProgramAssociations(somSelf, pszAssoc,
       fFilter);
```

wpSetProgramAssociations Parameter - somSelf

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

wpSetProgramAssociations Parameter - pszAssoc

pszAssoc (PSZ) - input
Pointer to string of datafile association types or filters, separated by commas.

wpSetProgramAssociations Parameter - fFilter

fFilter (BOOL) - input
Filter flag.

TRUE	*pszAssoc* contains filters.
FALSE	*pszAssoc* contains types.

wpSetProgramAssociations Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetProgramAssociations - Parameters

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

pszAssoc (PSZ) - input
Pointer to string of datafile association types or filters, separated by commas.

fFilter (BOOL) - input
Filter flag.

TRUE	*pszAssoc* contains filters.
FALSE	*pszAssoc* contains types.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetProgramAssociations - Usage

This method can be called at any time.

wpSetProgramAssociations - How to Override

This method is generally not overridden.

wpSetProgramAssociations - Related Methods

Related Methods

- wpQueryProgramAssociations

wpSetProgramAssociations - Topics

Select an item:
Syntax
Parameters
Returns

How to Override
Usage
Related Methods
Glossary

wpSetQueueOptions

wpSetQueueOptions - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the printer object's queue options.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPrinter     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
ULONG          ulOptions;  /*  Flag indicating queue option settings. */
BOOL           rc;         /*  Success indicator. */

rc = _wpSetQueueOptions(somSelf, ulOptions);
```

wpSetQueueOptions Parameter - somSelf

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

wpSetQueueOptions Parameter - ulOptions

ulOptions (ULONG) - input
Flag indicating queue option settings.

The following flags can be ORed together:

PO_PRINTERSPECIFIC
If set, the printer object spools print jobs in PM_Q_RAW format. If cleared, the printer object spools print jobs in PM_Q_STD format.

PO_PRINTWHILESPOOLING
If set, printing is enabled while the job is spooling. If cleared, printing occurs only after the job is spooled.

PO_APPDEFAULT
If set, this printer object becomes the application's default printer object. If cleared, the application's default printer object is not changed.

PO_DIALOGBEFOREPRINT
: If set, a Job Properties dialog is displayed when a job is submitted to this printer object. If cleared, the Job Properties dialog is not displayed.

wpSetQueueOptions Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
: Successful completion.

FALSE
: Error occurred.

wpSetQueueOptions - Parameters

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

ulOptions (ULONG) - input
Flag indicating queue option settings.

The following flags can be ORed together:

PO_PRINTERSPECIFIC
: If set, the printer object spools print jobs in PM_Q_RAW format. If cleared, the printer object spools print jobs in PM_Q_STD format.

PO_PRINTWHILESPOOLING
: If set, printing is enabled while the job is spooling. If cleared, printing occurs only after the job is spooled.

PO_APPDEFAULT
: If set, this printer object becomes the application's default printer object. If cleared, the application's default printer object is not changed.

PO_DIALOGBEFOREPRINT
: If set, a Job Properties dialog is displayed when a job is submitted to this printer object. If cleared, the Job Properties dialog is not displayed.

rc (BOOL) - returns
Success indicator.

TRUE
: Successful completion.

FALSE
: Error occurred.

wpSetQueueOptions - Usage

This method can be called at any time.

wpSetQueueOptions - How to Override

This method is generally not overridden.

wpSetQueueOptions - Related Methods

Related Methods

- wpQueryQueueOptions

wpSetQueueOptions - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpSetRealName

wpSetRealName - Syntax

This instance method sets the physical name of a file-system object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PSZ               pszName;  /*  Pointer to a new file name. */
BOOL              rc;       /*  Success indicator. */

rc = _wpSetRealName(somSelf, pszName);
```

wpSetRealName Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetRealName Parameter - pszName

pszName (PSZ) - input
Pointer to a new file name.

This file cannot be fully qualified.

wpSetRealName Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetRealName - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pszName (PSZ) - input
Pointer to a new file name.

This file cannot be fully qualified.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetRealName - Remarks

In most cases, the file-system object's real name and title are identical. When a title is set that the file system containing the object cannot handle, the real name is different. In this situation, real name is set to be a truncated title. In the case where the real name and the title are different, the title is stored in the file's ".LONGNAME" extended attribute.

wpSetRealName - Usage

This method can be called at any time in order to set the physical name for a file-system object.

wpSetRealName - How to Override

This method is generally not overridden.

wpSetRealName - Related Methods

Related Methods

- wpQueryRealName

wpSetRealName - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpSetRefreshFlags

wpSetRefreshFlags - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the refresh flags for a file system object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
ULONG             ulRefreshFlags;  /*  New DIRTYBIT and FOUNDBIT settings. */
BOOL              rc;              /*  Success indicator */

rc = _wpSetRefreshFlags(somSelf, ulRefreshFlags);
```

wpSetRefreshFlags Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpSetRefreshFlags Parameter - ulRefreshFlags

ulRefreshFlags (ULONG) - input
New DIRTYBIT and FOUNDBIT settings.

DIRTYBIT	(0x80000000) Used for refreshing the views of pre-existing objects.
FOUNDBIT	(0x40000000) Used for refreshing the views of new and deleted objects.

wpSetRefreshFlags Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetRefreshFlags - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

ulRefreshFlags (ULONG) - input
New DIRTYBIT and FOUNDBIT settings.

DIRTYBIT	(0x80000000) Used for refreshing the views of pre-existing objects.
FOUNDBIT	(0x40000000) Used for refreshing the views of new and deleted objects.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetRefreshFlags - Remarks

You should only call the wpSetRefreshFlags method only at a time when your object class is controlling the process of refreshing a folder. For example, if you have overridden the wpRefresh method for a subclass of WPFolder, you can set and query the refresh flags while your wpRefresh method is in control. You should not try to modify the refresh flags while the Workplace Shell's wpRefresh method is in control.

The refresh flags consist of a DIRTYBIT and a FOUNDBIT that are used to detect deleted files and new files when refreshing the contents of a folder.

For example, if you develop a subclass of wpFolder which contains objects that represent entries in a database (where the database may get updated independently from your objects), you might want to resynchronize your folder full of objects with the contents of the database. One possible technique follows:

1. Loop through all of the objects in the folder and turn on the DIRTYBIT and turn off the FOUNDBIT for all of your objects.
2. Loop through the database. For every entry in the database, find the corresponding object.
 a. If the object exists, turn on the FOUNDBIT for the object.
 b. If the object does not exist, create a new object with the FOUNDBIT turned on and the DIRTYBIT turned off.
3. Loop through the objects in the folder again. For any object that has the FOUNDBIT turned off, delete the object (since there is no longer a corresponding entry in the database). For any object that has the DIRTYBIT turned on, update the view with the current contents of the object and turn its DIRTYBIT off.

This kind of technique can be implemented using private flags, or by using the flags provided by the wpQueryRefreshFlags and wpSetRefreshFlags methods.

wpSetRefreshFlags - Usage

You can call the wpSetRefreshFlags method at any time to set the refresh flags for a file system object.

wpSetRefreshFlags - How to Override

This method is not generally overridden.

wpSetRefreshFlags - Related Methods

Related Methods

- wpQueryRefreshFlags

wpSetRefreshFlags - Example Code

This example shows how you can use the refresh flags to synchronize the contents of a folder with the contents of a data base. somSelf is the pointer to the folder containing the objects corresponding to the items in the data base.

```
MyObject    *Object,*NextObject;

/* Mark all existing objects as not found  */
Object = (MyObject*)_wpQueryContent(somSelf,NULL,QC_FIRST);
while (Object)
{
   _wpSetRefreshFlags(Object,0);
   Object = (MyObject*)_wpQueryContent(somSelf,Object,QC_NEXT);
}

/* Loop through the data base. For each item in the data base, find the
 * corresponding object and call the wpSetRefreshFlags method for the
 * object to set FOUNDBIT on. For any item that does not have a
 * corresponding object, create the object and call the wpSetRefreshFlags
 * method for the new object to set the FOUNDBIT on.   */

/***** The code for the above described loop goes here *****/
```

```
    /* Remove all objects corresponding to data base entries that have been
     * deleted from the data base */
    Object = (MyObject*)_wpQueryContent(somSelf,NULL,QC_FIRST);
    while (Object)
    {
       NextObject = (MyObject*)_wpQueryContent(somSelf,Object,QC_NEXT);
       ulFlags = _wpQueryRefreshFlags(Object);
       if (!(ulFlags & FOUNDBIT))
       {
          Object->wpDelete(0);
       }
       Object = NextObject;
    }
```

wpSetRefreshFlags - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetRefreshRate

wpSetRefreshRate - Syntax

This method is specific to version 2.1, or higher, of the OS/2 operating system.

This instance method sets the status window update rate for the power object when the automatic refresh option is enabled.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPower     *somSelf;        /*  Pointer to the object on which the method is being invoked. */
ULONG        ulRefreshRate;  /*  Contains the refresh rate value in minutes. */
BOOL         rc;             /*  Success indicator. */

rc = _wpSetRefreshRate(somSelf, ulRefreshRate);
```

wpSetRefreshRate Parameter - somSelf

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

wpSetRefreshRate Parameter - ulRefreshRate

ulRefreshRate (ULONG) - input
Contains the refresh rate value in minutes.

The range is 1-30.

wpSetRefreshRate Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion
FALSE	Failure, refresh rate is out of range.

wpSetRefreshRate - Parameters

somSelf (WPPower *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPower.

ulRefreshRate (ULONG) - input
Contains the refresh rate value in minutes.

The range is 1-30.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion
FALSE	Failure, refresh rate is out of range.

wpSetRefreshRate - Related Methods

Related Methods

- wpAddPowerPage
- wpAddPowerViewPage
- wpChangePowerState
- wpQueryAutoRefresh
- wpQueryPowerConfirmation
- wpQueryPowerManagement
- wpQueryRefreshRate
- wpSetAutoRefresh

- wpSetPowerConfirmation
- wpSetPowerManagement
- wpSetRefreshRate

wpSetRefreshRate - Topics

Select an item:
Syntax
Parameters
Returns
Related Methods
Glossary

wpSetRemoteOptions

wpSetRemoteOptions - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the job view options for a printer object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPrinter     *somSelf;            /*  Pointer to the object on which the method is being invoked. */
ULONG          ulRefreshInterval;  /*  Time interval, in seconds, when the printer object is refreshed. */
ULONG          flAllJobs;          /*  Flag indicating which jobs to display. */
BOOL           rc;                 /*  Success indicator. */

rc = _wpSetRemoteOptions(somSelf, ulRefreshInterval,
       flAllJobs);
```

wpSetRemoteOptions Parameter - somSelf

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

wpSetRemoteOptions Parameter - ulRefreshInterval

ulRefreshInterval (ULONG) - input
Time interval, in seconds, when the printer object is refreshed.

wpSetRemoteOptions Parameter - flAllJobs

flAllJobs (ULONG) - input
Flag indicating which jobs to display.

TRUE
Display all jobs waiting in the network queue.
FALSE
Display only the current user's jobs.

wpSetRemoteOptions Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetRemoteOptions - Parameters

somSelf (WPPrinter *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPrinter.

ulRefreshInterval (ULONG) - input
Time interval, in seconds, when the printer object is refreshed.

flAllJobs (ULONG) - input
Flag indicating which jobs to display.

TRUE
Display all jobs waiting in the network queue.
FALSE
Display only the current user's jobs.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetRemoteOptions - Usage

This method can be called at any time.

wpSetRemoteOptions - How to Override

This method is generally not overridden.

wpSetRemoteOptions - Related Methods

Related Methods

- wpQueryRemoteOptions

wpSetRemoteOptions - Example Code

```
BOOL        rc;
ULONG       ulOrigRefreshSeconds = 0;
ULONG       ulOrigAllJobsDisplay = 0;

ULONG       ulRefreshSeconds = 5;        /* Refresh interval of 5 seconds */
ULONG       ulAllJobsDisplay = TRUE;     /* Display all jobs              */

rc = _wpQueryRemoteOptions( somSelf
                          , &ulOrigRefreshSeconds
                          , &ulOrigAllJobsDisplay);

somPrintf("Original Refresh = %u, Jobs Display = %u.\n",
           ulOrigRefreshSeconds,ulOrigAllJobsDisplay);

rc = _wpSetRemoteOptions( somSelf
                        , ulRefreshSeconds
                        , ulAllJobsDisplay);
```

wpSetRemoteOptions - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetShadowTextColor

wpSetShadowTextColor - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the color to be used for the shadow text for the specified view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;        /*  Pointer to the object on which the method is being invoked. */
ULONG         ulColor;        /*  The RGB color value or USE_GLOBAL_COLOR. */
ULONG         ulView;         /*  Specifies which view to query. */
BOOL          fRefreshViews;  /*  Refresh open views flag. */

wpSetShadowTextColor(somSelf, ulColor, ulView,
       fRefreshViews);
```

wpSetShadowTextColor Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpSetShadowTextColor Parameter - ulColor

ulColor (ULONG) - input
The RGB color value or USE_GLOBAL_COLOR.

The RGB value must be presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

wpSetShadowTextColor Parameter - ulView

ulView (ULONG) - input
Specifies which view to query.

OPEN_CONTENTS
Opens the Contents view.
OPEN_DEFAULT
Opens the default view (same as double-clicking).
OPEN_DETAILS
Opens the Details view.
OPEN_HELP

Displays the Help panel.

OPEN_RUNNING
Executes the object.

OPEN_SETTINGS
Opens the Settings notebook.

OPEN_TREE
Opens the Tree view.

OPEN_USER
Class-specific views have a greater value than this.

wpSetShadowTextColor Parameter - fRefreshViews

fRefreshViews (BOOL) - input
Refresh open views flag.

TRUE	Refreshes all open views.
FALSE	Does not refresh the open views.

wpSetShadowTextColor - Return Value

There is no return value for this method.

wpSetShadowTextColor - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulColor (ULONG) - input
The RGB color value or USE_GLOBAL_COLOR.

The RGB value must be presented as a 6-digit hex value in the format 0xRRGGBB where RR, GG, and BB are the red, green, and blue values ranging between 0x00 and 0xFF (0-255).

ulView (ULONG) - input
Specifies which view to query.

OPEN_CONTENTS
Opens the Contents view.

OPEN_DEFAULT
Opens the default view (same as double-clicking).

OPEN_DETAILS
Opens the Details view.

OPEN_HELP
Displays the Help panel.

OPEN_RUNNING
Executes the object.

OPEN_SETTINGS
Opens the Settings notebook.

OPEN_TREE
Opens the Tree view.

OPEN_USER
Class-specific views have a greater value than this.

fRefreshViews (BOOL) - input
Refresh open views flag.

TRUE	Refreshes all open views.
FALSE	Does not refresh the open views.

There is no return value for this method.

wpSetShadowTextColor - Usage

This method can be called at any time in order to set the text color for the titles displayed for shadow icons in Icon, Tree, or Details View of a folder.

wpSetShadowTextColor - How to Override

This method is generally not overridden.

wpSetShadowTextColor - Related Methods

Related Methods

- wpQueryIconTextBackgroundColor
- wpQueryIconTextColor
- wpQueryIconTextVisibility
- wpQueryShadowTextColor
- wpSetIconTextBackgroundColor
- wpSetIconTextColor
- wpSetIconTextVisibility

wpSetShadowTextColor - Example Code

This example lets you set the icon text color for shadow icons in the Icon View for the current folder to BLUE.

```
_wpSetShadowTextColor(somSelf,0x0000FF);
```

wpSetShadowTextColor - Topics

Select an item:
Syntax
Parameters
Returns
How to Override

Usage
Example Code
Related Methods
Glossary

wpSetShadowTitle

wpSetShadowTitle - Syntax

This instance method is called to set the title on the shadow without affecting the title on the object it is shadowing.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPShadow     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
PSZ           pszNewTitle;  /*  Pointer to a title. */
BOOL          rc;           /*  Success indicator. */

rc = _wpSetShadowTitle(somSelf, pszNewTitle);
```

wpSetShadowTitle Parameter - somSelf

somSelf (WPShadow *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPShadow.

wpSetShadowTitle Parameter - pszNewTitle

pszNewTitle (PSZ) - input
Pointer to a title.

wpSetShadowTitle Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetShadowTitle - Parameters

somSelf (WPShadow *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPShadow.

pszNewTitle (PSZ) - input
Pointer to a title.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetShadowTitle - Usage

This method can be called at any time in order to set a title on a shadow object without affecting the title on the object it is shadowing.

wpSetShadowTitle - How to Override

This method is generally not overridden.

wpSetShadowTitle - Related Methods

Related Methods

- wpSetTitle

wpSetShadowTitle - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Related Methods
Glossary

wpSetSortAttribAvailable

wpSetSortAttribAvailable - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the availability of sort attributes for the current folder.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
ULONG         index;      /*  Column index for the current details view. */
BOOL          Available;  /*  Flag indicating whether to show or hide the sort attributes. */
BOOL          rc;         /*  Success indicator. */

rc = _wpSetSortAttribAvailable(somSelf, index,
       Available);
```

wpSetSortAttribAvailable Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpSetSortAttribAvailable Parameter - index

index (ULONG) - input
Column index for the current details view.

wpSetSortAttribAvailable Parameter - Available

Available (BOOL) - input
Flag indicating whether to show or hide the sort attributes.

Possible values are described in the following list:

TRUE
Show sort attributes.

FALSE
Hide sort attributes.

wpSetSortAttribAvailable Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetSortAttribAvailable - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

index (ULONG) - input
Column index for the current details view.

Available (BOOL) - input
Flag indicating whether to show or hide the sort attributes.

Possible values are described in the following list:

TRUE
Show sort attributes.
FALSE
Hide sort attributes.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetSortAttribAvailable - How to Override

This method is generally not overridden.

wpSetSortAttribAvailable - Related Methods

Related Methods

- wpIsSortAttribAvailable

wpSetSortAttribAvailable - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Related Methods
Glossary

wpSetStyle

wpSetStyle - Syntax

This instance method is called to allow an object to set its current object style.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
ULONG         ulNewStyle;  /*  Object styles. */
BOOL          rc;          /*  Success indicator. */

rc = _wpSetStyle(somSelf, ulNewStyle);
```

wpSetStyle Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetStyle Parameter - ulNewStyle

ulNewStyle (ULONG) - input
Object styles.

OBJSTYLE_CUSTOMICON
The icon is destroyed when the object goes dormant.
OBJSTYLE_NOCOPY
This object cannot be copied.
OBJSTYLE_NODELETE
This object cannot be deleted.
OBJSTYLE_NODRAG
This object cannot be dragged.
OBJSTYLE_NODROPON
No other object can be dropped on this object; however, this object can be dragged and dropped on other objects.

OBJSTYLE_NOLINK
This object cannot have a shadow created.
OBJSTYLE_NOMOVE
This object cannot move.
OBJSTYLE_NOPRINT
This object cannot be printed.
OBJSTYLE_NORENAME
This object cannot be renamed.
OBJSTYLE_NOSETTINGS
This object cannot be set.
OBJSTYLE_NOTVISIBLE
This object is not visible.
OBJSTYLE_TEMPLATE
This object is a template.
OBJSTYLE_LOCKEDINPLACE
Lock this object in place.

wpSetStyle Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetStyle - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

ulNewStyle (ULONG) - input
Object styles.

OBJSTYLE_CUSTOMICON
The icon is destroyed when the object goes dormant.
OBJSTYLE_NOCOPY
This object cannot be copied.
OBJSTYLE_NODELETE
This object cannot be deleted.
OBJSTYLE_NODRAG
This object cannot be dragged.
OBJSTYLE_NODROPON
No other object can be dropped on this object; however, this object can be dragged and dropped on other objects.
OBJSTYLE_NOLINK
This object cannot have a shadow created.
OBJSTYLE_NOMOVE
This object cannot move.
OBJSTYLE_NOPRINT
This object cannot be printed.
OBJSTYLE_NORENAME
This object cannot be renamed.
OBJSTYLE_NOSETTINGS
This object cannot be set.
OBJSTYLE_NOTVISIBLE
This object is not visible.
OBJSTYLE_TEMPLATE
This object is a template.
OBJSTYLE_LOCKEDINPLACE
Lock this object in place.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetStyle - Usage

This method can be called at any time in order to change an object's style.

wpSetStyle - How to Override

This method is generally not overridden.

wpSetStyle - Related Methods

Related Methods

- wpclsQueryStyle
- wpQueryStyle
- wpSetStyle

wpSetStyle - Example Code

This example removes the template style from the object so it can be deleted.

```
_wpSetStyle( objMyPrt
           , (_wpQueryStyle( objMyPrt ) & ~OBJSTYLE_TEMPLATE ) );

somPrintf("About to delete MyPrt Object\n");

_wpDelete( objMyPrt     /* Object to delete */
         , 0);          /* No confirmations on the delete */
```

wpSetStyle - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetTaskRec

wpSetTaskRec - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method adds, replaces, or deletes a task data block.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
PTASKREC      pNew;     /*  New task link to be added or replaced. */
PTASKREC      pOld;     /*  Old task link to be deleted or replaced. */
BOOL32        rc;       /*  Success indicator */

rc = _wpSetTaskRec(somSelf, pNew, pOld);
```

wpSetTaskRec Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetTaskRec Parameter - pNew

pNew (PTASKREC) - input
New task link to be added or replaced.

wpSetTaskRec Parameter - pOld

pOld (PTASKREC) - input
Old task link to be deleted or replaced.

wpSetTaskRec Return Value - rc

rc (BOOL32) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetTaskRec - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pNew (PTASKREC) - input
New task link to be added or replaced.

pOld (PTASKREC) - input
Old task link to be deleted or replaced.

rc (BOOL32) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetTaskRec - Remarks

The task records are used by the Workplace Shell to keep track of work being done on another thread (such as moving and copying objects).

wpSetTaskRec - How to Override

This method is not generally overridden.

wpSetTaskRec - Related Methods

Related Methods

- wpFindTaskRec

wpSetTaskRec - Topics

Select an item:
Syntax

Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpSetTitle

wpSetTitle - Syntax

This instance method is called to allow the object to set its current title.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;      /*  Pointer to the object on which the method is being invoked. */
PSZ           pszNewTitle;  /*  Pointer to the title of the object. */
BOOL          rc;           /*  Success indicator. */

rc = _wpSetTitle(somSelf, pszNewTitle);
```

wpSetTitle Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetTitle Parameter - pszNewTitle

pszNewTitle (PSZ) - input
Pointer to the title of the object.

A pointer to a zero-terminated string that contains the title of the object.

wpSetTitle Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetTitle - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pszNewTitle (PSZ) - input
Pointer to the title of the object.

A pointer to a zero-terminated string that contains the title of the object.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetTitle - Remarks

Valid titles must be less than CCHMAXPATHCOMP characters in length - currently defined as 256.

wpSetTitle - Usage

This method can be called at any time in order to set an object's title.

wpSetTitle - How to Override

This method is generally not overridden.

wpSetTitle - Related Methods

Related Methods

- wpQueryTitle
- wpclsQueryTitle
- wpSetTitle

wpSetTitle - Example Code

In this example, if templateObject is indeed a template, a new object is created from it.

```
   if( _wpQueryStyle( templateObject ) & OBJSTYLE_TEMPLATE )
   {
      destFolder = _wpclsQueryFolder(_WPFolder,"<WP_DESKTOP>",TRUE);

      MyPrinterObject = _wpCreateFromTemplate( templateObject
                                    , destFolder
                                    , FALSE );
      if (MyPrinterObject != NULLHANDLE) {
        _wpSetTitle( dummyPrinterObject, "My Printer Object" );
      } /* endif */

   }
```

wpSetTitle - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetTitleAndRenameFile

wpSetTitleAndRenameFile - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method sets the current title of an object and renames the file to match the title.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
PSZ               pszNewTitle;     /*  Pointer to new the title of object. */
ULONG             fConfirmations;  /*  Confirmation flag.          . */
BOOL              rc;              /*  Success indicator. */

rc = _wpSetTitleAndRenameFile(somSelf, pszNewTitle,
       fConfirmations);
```

wpSetTitleAndRenameFile Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpSetTitleAndRenameFile Parameter - pszNewTitle

pszNewTitle (PSZ) - input
Pointer to new the title of object.

A pointer to a zero terminated string which contains the new title of the object.

wpSetTitleAndRenameFile Parameter - fConfirmations

fConfirmations (ULONG) - input
Confirmation flag. .

Possible values are described in the following list:

CONFIRM_RENAMEFILEWITHEXT
Prompt the user to confirm renaming the file to match the title.

CONFIRM_KEEPASSOC
Prompt for confirmation when renaming files with extensions that are used to associate a file with an application.

For example, if the user has an association between an editor and the file extension DOC, and the user renames a DOC file to have a TXT extension, the file is no longer associated with the editor. The user should be prompted to confirm the name change.

0
Do not prompt the user to confirm renaming the file to match the title.

wpSetTitleAndRenameFile Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetTitleAndRenameFile - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

pszNewTitle (PSZ) - input
Pointer to new the title of object.

A pointer to a zero terminated string which contains the new title of the object.

fConfirmations (ULONG) - input
Confirmation flag. .

Possible values are described in the following list:

CONFIRM_RENAMEFILEWITHEXT
Prompt the user to confirm renaming the file to match the title.

CONFIRM_KEEPASSOC
Prompt for confirmation when renaming files with extensions that are used to associate a file with an application.

For example, if the user has an association between an editor and the file extension DOC, and the user renames a DOC file to have a TXT extension, the file is no longer associated with the editor. The user should be prompted to confirm the name change.

0
Do not prompt the user to confirm renaming the file to match the title.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetTitleAndRenameFile - Remarks

wpSetTitle is prefered for setting an object's title unless you want to override the confirmations; for example, if you want to rename a file and change its title without prompting the user.

wpSetTitleAndRenameFile - Usage

This method can be called at any time.

wpSetTitleAndRenameFile - How to Override

This method is generally not overridden.

wpSetTitleAndRenameFile - Related Methods

Related Methods

- wpConfirmKeepAssoc
- wpSetTitle

wpSetTitleAndRenameFile - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpSetType

wpSetType - Syntax

This instance method is called to allow the object to change the type of its file.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
PSZ               pszTypes;  /*  Pointer to buffer containing type to set. */
PFEA2LIST         pfeal;     /*  Reserved - must be set to 0. */
BOOL              rc;        /*  Success indicator. */

rc = _wpSetType(somSelf, pszTypes, pfeal);
```

wpSetType Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpSetType Parameter - pszTypes

pszTypes (PSZ) - input
Pointer to buffer containing type to set.

This string can contain a list of types delineated by a line-feed character; for example,

```
psztypes="Plain Text\nC code"
```

wpSetType Parameter - pfeal

pfeal (PFEA2LIST) - input
Reserved - must be set to 0.

wpSetType Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetType - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

pszTypes (PSZ) - input
Pointer to buffer containing type to set.

This string can contain a list of types delineated by a line-feed character; for example,

```
psztypes="Plain Text\nC code"
```

pfeal (PFEA2LIST) - input
Reserved - must be set to 0.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetType - Remarks

This method will cause the file's .TYPE extended attribute to be set.

wpSetType - Usage

This method can be called at any time in order to set the type on the file object.

wpSetType - How to Override

This method is generally not overridden.

wpSetType - Example Code

This example initializes the type of the instance of the Browse_O_Matic object to DLL.

```
SOM_Scope BOOL   SOMLINK Browse_O_Maticwps_wpSetup(Browse_O_Matic *somSelf,
                 PSZ pszSetupString)
{
    /* Browse_O_MaticData *somThis = Browse_O_MaticGetData(somSelf); */
    Browse_O_MaticMethodDebug("Browse_O_Matic","Browse_O_Maticwps_wpSetup");

    _wpSetType( somSelf, DRT_DLL, NULL);

    /*********************************************/
    /* Allow mulitple secondary views of the object */
    /*********************************************/

    _wpSetConcurrentView( somSelf, CCVIEW_ON);

    return (parent_wpSetup(somSelf,pszSetupString));
}
```

wpSetType - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Glossary

wpSetup

wpSetup - Syntax

This instance method is called to allow the newly created object to initialize itself based on an input setup string.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
PSZ           pszSetupString;  /*  Class-specific setup parameters for an object. */
BOOL          rc;              /*  Success indicator. */

rc = _wpSetup(somSelf, pszSetupString);
```

wpSetup Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetup Parameter - pszSetupString

pszSetupString (PSZ) - input
Class-specific setup parameters for an object.

wpSetup Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetup - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pszSetupString (PSZ) - input
Class-specific setup parameters for an object.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetup - Remarks

If wpSetup returns FALSE, the creation of the object is terminated.

The *pszSetupString* contains a series of "keyname=value" pairs that change the behavior of the object. Multiple "keyname=value" pairs are separated by semicolons. For example:

```
"key=value;key2=value1,value2;"
```

If you want a literal semicolon inside one of your fields you must type the following:

^;	A literal semicolon.

Each object class documents the keynames and the parameters it expects to see immediately following. Note that all parameters have safe defaults, so it is never required to pass parameters to an object.

For a listing of setup strings, see the individual object classes.

wpSetup - Usage

This method is generally called only by the system during the processing of wpclsNew (wpSetupOnce), WinCreateObject, and WinSetObjectData.

For more information about WinCreateObject and WinSetObjectData, see the *Presentation Manager Programming Reference* .

wpSetup - How to Override

This method is overridden by classes that introduce their own KEYNAMES.

wpSetup - Related Methods

Related Methods

- wpScanSetupString
- wpSetupOnce
- wpclsNew

wpSetup - Example Code

This example overrides _wpSetup, specifies setup strings, does some initialization, and writes some default data to the object's real filename. This will provide something to look at if the user drag/drops on an editor or selects the open/editor view.

```
SOM_Scope BOOL   SOMLINK myf_wpSetup(MYFILE *somSelf,
                 PSZ pszSetupString)
{
    MYFILEData *somThis = MYFILEGetData(somSelf);
    ULONG  cbBytesWritten;            /* Number of bytes written */
```

```
    APIRET rc;                          /* Return code */
    BOOL   fSuccess;                    /* Success flag */
    HFILE  hf;                          /* File handle */
    ULONG  ulAction;                    /* Action taken by DosOpen */
    CHAR   szObjectFilename[CCHMAXPATH];    /* Buffer for wpQueryRealName() */
    ULONG  cb  = sizeof(szObjectFilename);  /* Size of object */
    PSZ    pszDefaultText;              /* Default text */
    BOOL   rcParentCall;                /* Result of parent's method */
    CHAR   szValue[CCHMAXPATH+1];
    ULONG  cbBuffer;

    MYFILEMethodDebug("MYFILE","myf_wpSetup");

   /*   When the object is created from scratch, put some default
        text into the file on the hard disk                          */

   fSuccess =
   _wpQueryRealName(                /* query full-pathname of object's file */
      somSelf,                                  /* pointer to this object */
      szObjectFilename,                                 /* return buffer */
      &cb,                                              /* sizeof buffer */
      TRUE);                            /* request fully qualified pathname? */


   if (fSuccess)
   {
      rc = DosOpen(szObjectFilename, &hf, &ulAction, 20L, FILE_NORMAL,
                   FILE_OPEN | FILE_CREATE,
                   OPEN_ACCESS_READWRITE | OPEN_SHARE_DENYNONE,
                   NULL);

      if (rc == NO_ERROR) {

         pszDefaultText = _clsQueryDefaultFileText(_MYFILE);
         DosWrite(hf, pszDefaultText, strlen(pszDefaultText), &cbBytesWritten);
         DosClose(hf);
      } /* endif */

/*
 *  New Setup Strings parsed and processed by this override:
 *
 *             SOUND=(YES/NO)    YES will invoke _soundAlarm method
 *                               NO  will do nothing    */
   } else {
      _wpclsSetError( somSelf, MYFILE_QRN_FAILED );
   } /* endif (fSuccess) */

   rcParentCall = parent_wpSetup(somSelf,pszSetupString);

            /* Process setup strings we understand */

   cbBuffer = CCHMAXPATH;
   if ( _wpScanSetupString(somSelf, pszSetupString, "SOUND",
                           szValue,&cbBuffer) )
   {
      if ((szValue[0] == 'Y') && (szValue[1] == 'E') && (szValue[2] == 'S')) {
         _soundAlarm(somSelf);
      } /* endif */
   } /* endif */

  return( rcParentCall );
}
```

wpSetup - Topics

Select an item:

Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods

wpSetupCell

wpSetupCell - Syntax

This instance method is called to initialize a cell.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPPalette     *somSelf;    /*  Pointer to the object on which the method is being invoked. */
PVOID          pCellData;  /*  Pointer to the data to be stored. */
ULONG          cb;         /*  Size of the data to be stored. */
ULONG          x;          /*  X-coordinate of the cell to be set up. */
ULONG          y;          /*  Y-coordinate of the cell to be set up. */
BOOL           rc;         /*  Success indicator. */

rc = _wpSetupCell(somSelf, pCellData, cb,
       x, y);
```

wpSetupCell Parameter - somSelf

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

wpSetupCell Parameter - pCellData

pCellData (PVOID) - input
Pointer to the data to be stored.

wpSetupCell Parameter - cb

cb (ULONG) - input
Size of the data to be stored.

wpSetupCell Parameter - x

x (ULONG) - input
X-coordinate of the cell to be set up.

wpSetupCell Parameter - y

y (ULONG) - input
Y-coordinate of the cell to be set up.

wpSetupCell Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetupCell - Parameters

somSelf (WPPalette *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPPalette.

pCellData (PVOID) - input
Pointer to the data to be stored.

cb (ULONG) - input
Size of the data to be stored.

x (ULONG) - input
X-coordinate of the cell to be set up.

y (ULONG) - input
Y-coordinate of the cell to be set up.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSetupCell - Usage

This method can be called at any time in order to initialize a cell.

wpSetupCell - How to Override

This method is generally not overridden.

wpSetupCell - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Glossary

wpSetupOnce

wpSetupOnce - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to allow the newly created object to initialize itself.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;         /*  Pointer to the object on which the method is being invoked. */
PSZ           pszSetupString;  /*  Class specific setup parameters for an object. */
BOOL          rc;              /*  Success indicator. */

rc = _wpSetupOnce(somSelf, pszSetupString);
```

wpSetupOnce Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSetupOnce Parameter - pszSetupString

pszSetupString (PSZ) - input
Class specific setup parameters for an object.

wpSetupOnce Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetupOnce - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

pszSetupString (PSZ) - input
Class specific setup parameters for an object.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

wpSetupOnce - Remarks

If wpSetupOnce returns FALSE, the creation of the object is terminated.

The WPObject version of wpSetupOnce calls wpSetup.

The *pszSetupString* contains a series of "keyname=value" pairs that change the behavior of the object. Multiple "keyname=value" pairs are separated by semicolons. For example:

```
"key=value;key2=value1,value2;"
```

If you want a literal semicolon inside one of your fields you must type the following:

^; A literal semicolon.

Each object class documents the keynames and the parameters it expects to see immediately following. Note that all parameters have safe

defaults, so it is never required to pass parameters to an object.

For a listing of setup strings, see the individual object classes.

wpSetupOnce - Usage

This method is generally only called by the system during the processing of wpclsNew and WinCreateObject.

For more information about WinCreateObject, see the *Presentation Manager Programming Reference* .

wpSetupOnce - How to Override

This method is overridden by classes which require initialization in addition to setup string processing.

wpSetupOnce - Related Methods

Related Methods

- wpSetup

wpSetupOnce - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpSetWarnBeforeStart (WPProgram)

wpSetWarnBeforeStart (WPProgram) - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the program reference's Warn Before Start flag. If this flag is set, put up a warning dialog before starting the program.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgram     *somSelf;           /*  Pointer to the object on which the method is being invoked. */
BOOL           fWarnBeforeStart;  /*  A warning dialog before starting the program. */
BOOL           rc;                /*  Success indicator */

rc = _wpSetWarnBeforeStart(somSelf, fWarnBeforeStart);
```

wpSetWarnBeforeStart (WPProgram) Parameter - somSelf

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

wpSetWarnBeforeStart (WPProgram) Parameter - fWarnBeforeStar

fWarnBeforeStart (BOOL) - input
A warning dialog before starting the program.

wpSetWarnBeforeStart (WPProgram) Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetWarnBeforeStart (WPProgram) - Parameters

somSelf (WPProgram *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgram.

fWarnBeforeStart (BOOL) - input
A warning dialog before starting the program.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetWarnBeforeStart (WPProgram) - Usage

This method can be called at any time to set the Warn Before Start option for a program reference object.

wpSetWarnBeforeStart (WPProgram) - How to Override

This method is not generally overridden.

wpSetWarnBeforeStart (WPProgram) - Related Methods

Related Methods

- wpQueryWarnBeforeStart (WPProgram)
- wpQueryWarnBeforeStart (WPProgramFile)
- wpSetWarnBeforeStart (WPProgram)

wpSetWarnBeforeStart (WPProgram) - Example Code

This example requests that the user be warned before starting the current program.

```
_wpSetWarnBeforeStart(somSelf,TRUE);
```

wpSetWarnBeforeStart (WPProgram) - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpSetWarnBeforeStart (WPProgramFile)

wpSetWarnBeforeStart (WPProgramFile) - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method sets the program reference's Warn Before Start flag. If this flag is set, put up a warning dialog before starting the program.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPProgramFile     *somSelf;           /*  Pointer to the object on which the method is being invoked. */
BOOL               rc;                /*  Success indicator */
BOOL               fWarnBeforeStart;  /*  A warning dialog before starting the program. */

fWarnBeforeStart = _wpSetWarnBeforeStart(
                    somSelf, rc);
```

wpSetWarnBeforeStart (WPProgramFile) Parameter - somSelf

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

wpSetWarnBeforeStart (WPProgramFile) Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpSetWarnBeforeStart (WPProgramFile) Parameter - fWarnBefore

fWarnBeforeStart (BOOL) - input
A warning dialog before starting the program.

wpSetWarnBeforeStart (WPProgramFile) - Parameters

somSelf (WPProgramFile *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPProgramFile.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

fWarnBeforeStart (BOOL) - input
A warning dialog before starting the program.

wpSetWarnBeforeStart (WPProgramFile) - Usage

This method can be called at any time to set the Warn Before Start option for a program file object.

wpSetWarnBeforeStart (WPProgramFile) - How to Override

This method is not generally overridden.

wpSetWarnBeforeStart (WPProgramFile) - Related Methods

Related Methods

- wpQueryWarnBeforeStart (WPProgram)
- wpQueryWarnBeforeStart (WPProgramFile)
- wpSetWarnBeforeStart (WPProgram)

wpSetWarnBeforeStart (WPProgramFile) - Example Code

This example requests that the user be warned before starting the current program.

```
_wpSetWarnBeforeStart(somSelf,TRUE);
```

wpSetWarnBeforeStart (WPProgramFile) - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Usage
Example Code
Related Methods
Glossary

wpStartJobAgain

wpStartJobAgain - Syntax

This instance method is called to start printing a job object again.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPJob     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL       rc;       /*  Success indicator. */

rc = _wpStartJobAgain(somSelf);
```

wpStartJobAgain Parameter - somSelf

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

wpStartJobAgain Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpStartJobAgain - Parameters

somSelf (WPJob *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPJob.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpStartJobAgain - How to Override

This method is generally not overridden.

wpStartJobAgain - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Glossary

wpStoreFldrRunObjs

wpStoreFldrRunObjs - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method is called when a work area folder is closed. It saves a list of the objects within the work area folder that are open when the work area folder is closed. It then closes all open views of these objects.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         ulType;   /*  Folder type flag. */
BOOL          rc;       /*  Success indicator */

rc = _wpStoreFldrRunObjs(somSelf, ulType);
```

wpStoreFldrRunObjs Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpStoreFldrRunObjs Parameter - ulType

ulType (ULONG) - input
Folder type flag.

FOI_WORKAREA	Designates a work area folder.

wpStoreFldrRunObjs Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpStoreFldrRunObjs - Parameters

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

ulType (ULONG) - input
Folder type flag.

FOI_WORKAREA	Designates a work area folder.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpStoreFldrRunObjs - Remarks

The system calls this method while closing a work area folder to save the list of objects that are open so they can be restored when the folder is reopened.

This method is only called on folders with the FOI_WORKAREA flag set when the folder is closed.

wpStoreFldrRunObjs - Usage

This method can be called at any time to save the list of open objects in a folder.

wpStoreFldrRunObjs - How to Override

This method is not generally overridden.

wpStoreFldrRunObjs - Related Methods

Related Methods

- wpHideFldrRunObjs
- wpRestoreFldrRunObjs

wpStoreFldrRunObjs - Example Code

This example saves the state of all open objects in a work area folder and closes all open views of the folder.

```
_wpStoreFldrRunObjs(WorkAreaFolder,CLOSE_FOLDER);
```

wpStoreFldrRunObjs - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpStoreIconPosData

wpStoreIconPosData - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method stores the ICONPOS data passed to this method on the folder object's .ICONPOS extended attribute (EA).

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFolder     *somSelf;   /*  Pointer to the object on which the method is being invoked. */
PICONPOS      pIconPos;  /*  Pointer to the ICONPOS data. */
ULONG         cbSize;    /*  Size of the ICONPOS data block. */
BOOL          rc;        /*  Success indicator */

rc = _wpStoreIconPosData(somSelf, pIconPos,
       cbSize);
```

wpStoreIconPosData Parameter - somSelf

somSelf (WPFolder *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

wpStoreIconPosData Parameter - pIconPos

pIconPos (PICONPOS) - input
Pointer to the ICONPOS data.

wpStoreIconPosData Parameter - cbSize

cbSize (ULONG) - input
Size of the ICONPOS data block.

wpStoreIconPosData Return Value - rc

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpStoreIconPosData - Parameters

somSelf (WPFolder *) - input

Pointer to the object on which the method is being invoked.

Points to an object of class WPFolder.

pIconPos (PICONPOS) - input
Pointer to the ICONPOS data.

cbSize (ULONG) - input
Size of the ICONPOS data block.

rc (BOOL) - returns
Success indicator

TRUE	Successful completion.
FALSE	Error occurred.

wpStoreIconPosData - Remarks

This method is invoked when the folder's open content view is closed.

wpStoreIconPosData - How to Override

This method is not generally overridden.

wpStoreIconPosData - Related Methods

Related Methods

- wpFreeIconPosData
- wpInitIconPosData
- wpQueryDefaultIconPos
- wpQueryIconPosition
- wpQueryNextIconPos
- wpSetDefaultIconPos
- wpSetNextIconPos

wpStoreIconPosData - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Related Methods
Glossary

wpSwitchTo

wpSwitchTo - Syntax

This instance method is called to allow the object to give focus to the specified open view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG         View;     /*  Flag indicating the open view to which focus is to be given. */
BOOL          rc;       /*  Success indicator. */

rc = _wpSwitchTo(somSelf, View);
```

wpSwitchTo Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpSwitchTo Parameter - View

View (ULONG) - input
Flag indicating the open view to which focus is to be given.

OPEN_CONTENTS
Open contents view.
OPEN_DEFAULT
Open default view (same as double-click).
OPEN_DETAILS
Open details view.
OPEN_HELP
Display HelpPanel.
OPEN_RUNNING
Execute object.
OPEN_SETTINGS
Open Settings notebook.
OPEN_TREE
Open tree view.
OPEN_USER
Class-specific views have a greater value than this.

wpSwitchTo Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSwitchTo - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

View (ULONG) - input
Flag indicating the open view to which focus is to be given.

OPEN_CONTENTS
: Open contents view.

OPEN_DEFAULT
: Open default view (same as double-click).

OPEN_DETAILS
: Open details view.

OPEN_HELP
: Display HelpPanel.

OPEN_RUNNING
: Execute object.

OPEN_SETTINGS
: Open Settings notebook.

OPEN_TREE
: Open tree view.

OPEN_USER
: Class-specific views have a greater value than this.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpSwitchTo - Remarks

The focus is given to the specified open view of the object if it exists. This is done by scanning the in-use list.

wpSwitchTo - Usage

This method can be called at any time in order to switch to an existing view of this object.

wpSwitchTo - How to Override

This method is generally not overridden.

wpSwitchTo - Related Methods

Related Methods

- wpOpen

wpSwitchTo - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpUnInitData

wpUnInitData - Syntax

This instance method is called to allow the object to free allocated resources.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */

wpUnInitData(somSelf);
```

wpUnInitData Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpUnInitData - Return Value

There is no return value for this method.

wpUnInitData - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

There is no return value for this method.

wpUnInitData - Remarks

The parent method must be called after you have completed your own processing.

wpUnInitData - Usage

This method is generally called only by the system when the object is made dormant. The object is made dormant when it is destroyed or when there are no open views and the folder containing the object are not open.

wpUnInitData - How to Override

This method is overridden to de-allocate resources allocated during the processing of wpInitData.

The parent method must be called after you have completed your own processing.

wpUnInitData - Related Methods

Related Methods

- wpInitData

wpUnInitData - Example Code

This example clears up memory that was allocated on wpInitData.

```
SOM_Scope void   SOMLINK myf_wpUnInitData(MYFILE *somSelf)
{
    MYFILEData *somThis = MYFILEGetData(somSelf);
    MYFILEMethodDebug("MYFILE","myf_wpUnInitData");

    if (_pszLastWorker)
    {
      _wpFreeMem(somSelf,
                 (PBYTE)_pszLastWorker);  /* Free this area */
    }

    parent_wpUnInitData(somSelf);  /* Call parent method last */

}
```

wpUnInitData - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

wpUnlockObject

wpUnlockObject - Syntax

This instance method decrements an object's lock count.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL          rc;       /*  Success indicator. */

rc = _wpUnlockObject(somSelf);
```

wpUnlockObject Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpUnlockObject Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpUnlockObject - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpUnlockObject - Remarks

wpUnlockObject unlocks the specified object. When an object is unlocked, it automatically goes into the dormant state when it no longer has open views and its container is no longer open.

wpUnlockObject - Usage

This method can be called at any time to decrement the object's lock count.

This method is used in conjunction with other methods that create new instances of objects. If these methods are called with the fLock flag set to TRUE, the new object is locked into the awake state until the wpUnlockObject method is called.

wpUnlockObject - How to Override

This method is generally not overridden.

wpUnlockObject - Related Methods

Related Methods

- wpCopyObject
- wpQueryContent
- wpclsCreateDefaultTemplates
- wpclsNew

wpUnlockObject - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Related Methods
Glossary

wpVerifyUpdateAccess

wpVerifyUpdateAccess - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method is called to verify the update access of the current file system.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPFileSystem     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
ULONG             rc;       /*  DosOpen error return code. */

rc = _wpVerifyUpdateAccess(somSelf);
```

wpVerifyUpdateAccess Parameter - somSelf

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

wpVerifyUpdateAccess Return Value - rc

rc (ULONG) - returns
DosOpen error return code.

Possible values are described in the following list:

NO_ERROR

ERROR_FILE_NOT_FOUND

ERROR_PATH_NOT_FOUND

ERROR_TOO_MANY_OPEN_FILES

ERROR_ACCESS_DENIED

ERROR_INVALID_ACCESS

ERROR_NOT_DOS_DISK

ERROR_SHARING_VIOLATION

ERROR_SHARING_BUFFER_EXCEEDED

ERROR_CANNOT_MAKE

ERROR_INVALID_PARAMETER

ERROR_DEVICE_IN_USE

ERROR_DRIVE_LOCKED

ERROR_OPEN_FAILED

ERROR_DISK_FULL

ERROR_FILENAME_EXCED_RANGE

ERROR_PIPE_BUSY

wpVerifyUpdateAccess - Parameters

somSelf (WPFileSystem *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPFileSystem.

rc (ULONG) - returns
DosOpen error return code.

Possible values are described in the following list:

NO_ERROR

ERROR_FILE_NOT_FOUND

ERROR_PATH_NOT_FOUND

ERROR_TOO_MANY_OPEN_FILES

ERROR_ACCESS_DENIED

ERROR_INVALID_ACCESS

ERROR_NOT_DOS_DISK

ERROR_SHARING_VIOLATION

ERROR_SHARING_BUFFER_EXCEEDED

ERROR_CANNOT_MAKE

ERROR_INVALID_PARAMETER

ERROR_DEVICE_IN_USE

ERROR_DRIVE_LOCKED

ERROR_OPEN_FAILED

ERROR_DISK_FULL

ERROR_FILENAME_EXCED_RANGE

ERROR_PIPE_BUSY

wpVerifyUpdateAccess - How to Override

This method is generally not overridden.

wpVerifyUpdateAccess - Topics

Select an item:
Syntax
Parameters
Returns
How to Override
Glossary

wpViewObject

wpViewObject - Syntax

This instance method either opens a view of the given object or resurfaces an existing view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
HWND          hwndCnr;  /*  Handle for the view of the object's container. */
ULONG         ulView;   /*  Flag indicating how to view the object. */
ULONG         param;    /*  Optional parameter. */
HWND          hwnd;     /*  Window handle of the view that has been opened or resurfaced. */

hwnd = _wpViewObject(somSelf, hwndCnr, ulView,
         param);
```

wpViewObject Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpViewObject Parameter - hwndCnr

hwndCnr (HWND) - input
Handle for the view of the object's container.

wpViewObject Parameter - ulView

ulView (ULONG) - input
Flag indicating how to view the object.

This parameter takes one of the following flags or any other defined open view for that class.

OPEN_CONTENTS
Open contents view for folders only.
OPEN_DEFAULT
Open default view (same as double-click).
OPEN_DETAILS
Open details view.
OPEN_HELP
Display HelpPanel.
OPEN_PALETTE
Palettes only.
OPEN_RUNNING
Execute object.
OPEN_SETTINGS
Open Settings notebook.
OPEN_TREE
Open tree view for folders only.
OPEN_USER
Class-specific views have a greater value than this.

wpViewObject Parameter - param

param (ULONG) - input
Optional parameter.

wpViewObject Return Value - hwnd

hwnd (HWND) - returns
Window handle of the view that has been opened or resurfaced.

A return value of NULLHANDLE indicates that an error occurred while trying to view the object.

wpViewObject - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

hwndCnr (HWND) - input
Handle for the view of the object's container.

ulView (ULONG) - input
Flag indicating how to view the object.

This parameter takes one of the following flags or any other defined open view for that class.

OPEN_CONTENTS
: Open contents view for folders only.

OPEN_DEFAULT
: Open default view (same as double-click).

OPEN_DETAILS
: Open details view.

OPEN_HELP
: Display HelpPanel.

OPEN_PALETTE
: Palettes only.

OPEN_RUNNING
: Execute object.

OPEN_SETTINGS
: Open Settings notebook.

OPEN_TREE
: Open tree view for folders only.

OPEN_USER
: Class-specific views have a greater value than this.

param (ULONG) - input
Optional parameter.

hwnd (HWND) - returns
Window handle of the view that has been opened or resurfaced.

A return value of NULLHANDLE indicates that an error occurred while trying to view the object.

wpViewObject - Remarks

Depending on whether the view of this object already exists and whether multiple concurrent views are set for this object, the parameters are passed on to either wpOpen to open a view of this object or to wpSwitchTo to surface an existing view.

It is recommended that you use this method instead of wpOpen.

wpViewObject - Usage

This method can be called at any time to view an object.

wpViewObject - How to Override

This method is generally not overridden.

wpViewObject - Related Methods

Related Methods

- wpOpen
- wpSwitchTo

wpViewObject - Example Code

This example opens Hex and Text views.

```
SOM_Scope BOOL   SOMLINK Browse_O_Maticwps_wpMenuItemSelected(Browse_O_Matic *somSelf,
                HWND hwndFrame,
                ULONG ulMenuId)
{
    /* Browse_O_MaticData *somThis = Browse_O_MaticGetData(somSelf); */
    Browse_O_MaticMethodDebug("Browse_O_Matic","Browse_O_Maticwps_wpMenuItemSelected");
    switch( ulMenuId ){
       case ID_OPEN_TEXT:
          _wpViewObject( somSelf, NULLHANDLE, ulMenuId, 0L );
          break;
       case ID_OPEN_HEX:
          _wpViewObject( somSelf, NULLHANDLE, ulMenuId, 0L );
          break;
       case ID_OPEN_PRODUCTINFO:
       case ID_PRODUCTINFOMENU:
          _wpViewObject( somSelf, NULLHANDLE, ulMenuId, 0L );

       default:
          return( parent_wpMenuItemSelected( somSelf, hwndFrame, ulMenuId) );
    }
    return( TRUE );
}
```

wpViewObject - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override

Usage
Example Code
Related Methods
Glossary

wpWaitForClose

wpWaitForClose - Syntax

This method is specific to version 3, or higher, of the OS/2 operating system.

This instance method pauses the application until the specified views are closed or the timeout value is reached.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPObject     *somSelf;     /*  Pointer to the object on which the method is being invoked. */
LHANDLE       lhView;      /*  Handle (hwnd or happ) of the view that must close before the application can cont
ULONG         ulViews;     /*  Type of views to wait for. */
ULONG         ulTimeOut;   /*  Time to wait for the views to close. */
BOOL          bAutoClose;  /*  Flag indicating whether to automatically close. */
ULONG         Success;     /*  Return value from WinWaitEventSem function. */

Success = _wpWaitForClose(somSelf, lhView,
            ulViews, ulTimeOut, bAutoClose);
```

wpWaitForClose Parameter - somSelf

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

wpWaitForClose Parameter - lhView

lhView (LHANDLE) - input
Handle (hwnd or happ) of the view that must close before the application can continue.

If a value of NULL is passed, this method waits for all views of the specified type (*ulViews*) to close.

wpWaitForClose Parameter - ulViews

ulViews (ULONG) - input
Type of views to wait for.

This value is only used if *hView* is NULL.

wpWaitForClose Parameter - ulTimeOut

ulTimeOut (ULONG) - input
Time to wait for the views to close.

Timeout in milliseconds. This is the maximum amount of time the user wants to allow the thread to be blocked.

This parameter can also have the following values:

SEM_IMMEDIATE_RETURN
Returns immediately without blocking the calling thread.
SEM_INDEFINITE_WAIT
Blocks the calling thread indefinitely.

wpWaitForClose Parameter - bAutoClose

bAutoClose (BOOL) - input
Flag indicating whether to automatically close.

This parameter should not be used as the standard method to close views. It exists ONLY to handle the possibility of a timing hole where all of the specified views are closed and another thread (probably due to user input) opens another view of the same type before wpWaitForClose returns.

Possible values are described in the following list:

TRUE
WM_Close is posted for the views that are not presently closing, or WinTerminateApp is called on the applications that are being waited on.

FALSE
Views that are not presently closing are not automatically closed.

wpWaitForClose Return Value - Success

Success (ULONG) - returns
Return value from WinWaitEventSem function.

This parameter can also have the following values:

0
The view was closed successfully or was already closed when this method processed.
-1
Error occurred.

wpWaitForClose - Parameters

somSelf (WPObject *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPObject.

lhView (LHANDLE) - input
Handle (hwnd or happ) of the view that must close before the application can continue.

If a value of NULL is passed, this method waits for all views of the specified type (*ulViews*) to close.

ulViews (ULONG) - input
Type of views to wait for.

This value is only used if *lhView* is NULL.

ulTimeOut (ULONG) - input
Time to wait for the views to close.

Timeout in milliseconds. This is the maximum amount of time the user wants to allow the thread to be blocked.

This parameter can also have the following values:

SEM_IMMEDIATE_RETURN
Returns immediately without blocking the calling thread.
SEM_INDEFINITE_WAIT
Blocks the calling thread indefinitely.

bAutoClose (BOOL) - input
Flag indicating whether to automatically close.

This parameter should not be used as the standard method to close views. It exists ONLY to handle the possibility of a timing hole where all of the specified views are closed and another thread (probably due to user input) opens another view of the same type before wpWaitForClose returns.

Possible values are described in the following list:

TRUE
WM_Close is posted for the views that are not presently closing, or WinTerminateApp is called on the applications that are being waited on.

FALSE
Views that are not presently closing are not automatically closed.

Success (ULONG) - returns
Return value from WinWaitEventSem function.

This parameter can also have the following values:

0
The view was closed successfully or was already closed when this method processed.
-1
Error occurred.

wpWaitForClose - Remarks

When the specified views appear to have finished closing on the Desktop it might seems that the method is hung. This is because closing the multithreading process may require additional time to complete.

If a view is in the process of closing, and you must wait for that view to finish closing before you can continue processing, this method can be called to act as an event semaphore. It can also be used to block the thread until a program exits when executed from a program object.

wpWaitForClose - Usage

This method can be called at any time.

wpWaitForClose - How to Override

This method is generally not overridden.

wpWaitForClose - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Glossary

wpWriteImageFile

wpWriteImageFile - Syntax

This method is specific to Version 4, or higher, of the OS/2 operating system.

This instance method writes the image data for the image file from the instance data.

```
#define INCL_WINWORKPLACE
#include <os2.h>

WPBitmap     *somSelf;  /*  Pointer to the object on which the method is being invoked. */
BOOL32        rc;       /*  Success indicator. */

rc = _wpWriteImageFile(somSelf);
```

wpWriteImageFile Parameter - somSelf

somSelf (WPBitmap *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPBitmap.

wpWriteImageFile Return Value - rc

rc (BOOL32) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpWriteImageFile - Parameters

somSelf (WPBitmap *) - input
Pointer to the object on which the method is being invoked.

Points to an object of class WPBitmap.

rc (BOOL32) - returns
Success indicator.

TRUE	Successful completion.
FALSE	Error occurred.

wpWriteImageFile - Remarks

In the implementation of the WPBitmap class in the Workplace Shell, the wpWriteImageFile method writes the current bitmap to the image file. For any other image type, the wpWriteImageFile method should convert the bitmap data to the appropriate image format and write the data out to the image file.

wpWriteImageFile - Usage

Various WPImageFile methods call the wpWriteImageFile method to change the contents of an image file.

This method converts the image data stored in the instance data from OS/2 Bitmap format to the format appropriate to the image class and writes it to the image data file.

wpWriteImageFile - How to Override

This method must be overridden by any subclass of WPImageFile.

wpWriteImageFile - Related Methods

Related Methods

- wpQueryBitmapData
- wpQueryBitmapHandle
- wpQueryBitmapInfoHeader
- wpSetBitmapData
- wpReadImageFile

wpWriteImageFile - Example Code

This example shows how the WPBitmap class writes the data to the image file.

```
SOM_Scope BOOL32 bmp_wpWriteImageFile(WPBitmap *somSelf)
{
    CHAR                szPath[CCHMAXPATH];
    HFILE               hfile = NULLHANDLE;
    ULONG               ulAction, ulBytesWritten = 0, ulSize=0;
    APIRET              rc;
    PBYTE               pBitmapData;
    ULONG               ulBitmapDataSize;

    /* Get the pointer to the bitmap data and its size  */
    pBitmapData = _wpQueryBitmapData(somSelf,&ulBitmapDataSize);

    /* Find the bitmap data file  */
    _wpQueryFilename(somSelf, szPath, TRUE);

    /* Write the file from the buffer  */
    rc = DosOpen(szPath,
                 &hfile,
                 &ulAction,
                 ulSize,
                 FILE_NORMAL,
                 FILE_OPEN,
                 OPEN_ACCESS_WRITEONLY | OPEN_SHARE_DENYNONE,
                 NULL);

    if (rc == 0)
    {
       rc = DosWrite(hfile,
                     pBitmapData,
                     ulBitmapDataSize,
                     &ulBytesWritten);

       DosClose(hfile);
       hfile = NULLHANDLE;
    }
    return (!rc && (ulBytesWritten == ulBitmapDataSize));
}
```

wpWriteImageFile - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
How to Override
Usage
Example Code
Related Methods
Glossary

Presentation Manager Functions Related to the Workplace Shell

This section contains an alphabetical list of the Presentation Manager (PM) functions which are available to the application for using and controlling Workplace Shell objects.

WinCopyObject

WinCopyObject - Syntax

This function is specific to OS/2 Version 3.0 or higher.

This function copies an object from its existing location to a specified new destination.

```
#define INCL_WINWORKPLACE
#include <os2.h>

HOBJECT    hObjectofObject;  /*  Handle of the Workplace Shell object being copied. */
HOBJECT    hObjectofDest;    /*  Handle of the destination folder into which hObjectofObject is to be copied. */
ULONG      ulReserved;       /*  Reserved value; must be 0. */
HOBJECT    rc;               /*  Handle of the newly created object. */

rc = WinCopyObject(hObjectofObject, hObjectofDest,
       ulReserved);
```

WinCopyObject Parameter - hObjectofObject

hObjectofObject (HOBJECT) - input
Handle of the Workplace Shell object being copied.

WinCopyObject Parameter - hObjectofDest

hObjectofDest (HOBJECT) - input
Handle of the destination folder into which *hObjectofObject* is to be copied.

WinCopyObject Parameter - ulReserved

ulReserved (ULONG) - input
Reserved value; must be 0.

WinCopyObject Return Value - rc

rc (HOBJECT) - returns
Handle of the newly created object.

A return value of NULLHANDLE indicates that either *hObjectofDest* is NULLHANDLE or an object with the same name as *hObjectofObject* exists in the destination folder.

WinCopyObject - Parameters

hObjectofObject (HOBJECT) - input
Handle of the Workplace Shell object being copied.

hObjectofDest (HOBJECT) - input
Handle of the destination folder into which *hObjectofObject* is to be copied.

ulReserved (ULONG) - input
Reserved value; must be 0.

rc (HOBJECT) - returns
Handle of the newly created object.

A return value of NULLHANDLE indicates that either *hObjectofDest* is NULLHANDLE or an object with the same name as *hObjectofObject* exists in the destination folder.

WinCopyObject - Remarks

Using HOBJECTs for .INI files or files in which an application uses a rename/save/delete sequence is not supported. Its REXX counterpart is SysCopyObject.

WinCopyObject - Errors

Possible returns from WinGetLastError

WPERR_INVALID_FLAGS (0x1719)
An invalid flag was specified.

WinCopyObject - Related Functions

Related Functions

- WinCreateObject
- WinDestroyObject
- WinMoveObject
- WinQueryObjectWindow

- WinSaveObject

WinCopyObject - Example Code

This example copies the drives object into the startup folder; the drives folder will be opened at the startup of the system.

```
#define INCL_WINWORKPLACE
#include "os2.h"

HOBJECT    hObjectofDest;
HOBJECT    hObjectofObject;
HOBJECT    hObjectofResult;

hObjectofObject = WinQueryObject("<WP_DRIVES>");
if (hObjectofObject != NULL)
{
  /* WinQueryObject of Drives was successful */
  hObjectofDest = WinQueryObject("<WP_START>");

  if (hObjectofDest != NULL)
  {
    /* WinQueryObject of Startup was successful */
    hObjectofResult = WinCopyObject(hObjectofObject, hObjectofDest, 0);

    if (hObjectofResult != NULL)
    {
      /* Drives Object was successfully copied to the Startup Folder */
    }
    else
    {
      /* Copy failed */
    }
  }
}
```

WinCopyObject - Topics

Select an item:
Syntax
Parameters
Returns
Errors
Remarks
Example Code
Related Functions
Glossary

WinCreateObject

WinCreateObject - Syntax

This function creates an instance of object class *pszClassName* , with title *pszTitle* , and places the icon and title in the location referred to by *pszLocation* .

```
#define INCL_WINWORKPLACE
#include <os2.h>

PSZ         pszClassName;    /*  Pointer to class name. */
PSZ         pszTitle;        /*  Pointer to initial title of object. */
PSZ         pszSetupString;  /*  Pointer to setup string. */
PSZ         pszLocation;     /*  Folder location. */
ULONG       ulFlags;         /*  Creation flags. */
HOBJECT     rc;              /*  Handle to the created object. */

rc = WinCreateObject(pszClassName, pszTitle,
       pszSetupString, pszLocation, ulFlags);
```

WinCreateObject Parameter - pszClassName

pszClassName (PSZ) - input
Pointer to class name.

A pointer to a zero-terminated string which contains the name of the class of which this object is a member.

WinCreateObject Parameter - pszTitle

pszTitle (PSZ) - input
Pointer to initial title of object.

A pointer to a zero-terminated string which contains the initial title of the object as it is to appear when displayed on the user interface underneath an icon or on the title bar of an open object.

WinCreateObject Parameter - pszSetupString

pszSetupString (PSZ) - input
Pointer to setup string.

See WPLaunchPad for a table of setup strings used to customize the Toolbar.

WinCreateObject Parameter - pszLocation

pszLocation (PSZ) - input
Folder location.

This value can be in any of the following formats:

- Predefined object ids of system folders.

"<WP_NOWHERE>"	The hidden folder.
"<LOCATION_DESKTOP>"	The currently active desktop.
"<WP_OS2SYS>"	The System folder.
"<WP_TEMPS>"	The Templates folder.
"<WP_CONFIG>"	The System Setup folder.
"<WP_START>"	The Startup folder.
"<WP_INFO>"	The Information folder.
"<WP_DRIVES>"	The Drives folder.

- Real name specified as a fully qualified path name.

WinCreateObject Parameter - ulFlags

ulFlags (ULONG) - input
Creation flags.

This parameter can have one of the following values:

CO_FAILIFEXISTS
No object will be created if an object with the given object ID already exists. This is the default.

CO_REPLACEIFEXISTS
If an object with the given ID already exists, the existing object should be replaced.

CO_UPDATEIFEXISTS
If an object with the given ID already exists, the existing object should be updated with the new information.

WinCreateObject Return Value - rc

rc (HOBJECT) - returns
Handle to the created object.

NULLHANDLE
Error occurred.
Other
A handle to the object created. This handle is persistent and can be used for the WinSetObjectData and WinDestroyObject function calls.

WinCreateObject - Parameters

pszClassName (PSZ) - input
Pointer to class name.

A pointer to a zero-terminated string which contains the name of the class of which this object is a member.

pszTitle (PSZ) - input
Pointer to initial title of object.

A pointer to a zero-terminated string which contains the initial title of the object as it is to appear when displayed on the user interface underneath an icon or on the title bar of an open object.

pszSetupString (PSZ) - input
Pointer to setup string.

See WPLaunchPad for a table of setup strings used to customize the Toolbar.

pszLocation (PSZ) - input
Folder location.

This value can be in any of the following formats:

- Predefined object ids of system folders.

"<WP_NOWHERE>"	The hidden folder.
"<LOCATION_DESKTOP>"	The currently active desktop.
"<WP_OS2SYS>"	The System folder.
"<WP_TEMPS>"	The Templates folder.
"<WP_CONFIG>"	The System Setup folder.
"<WP_START>"	The Startup folder.
"<WP_INFO>"	The Information folder.
"<WP_DRIVES>"	The Drives folder.

- Real name specified as a fully qualified path name.

ulFlags (ULONG) - input
Creation flags.

This parameter can have one of the following values:

CO_FAILIFEXISTS
No object will be created if an object with the given object ID already exists. This is the default.

CO_REPLACEIFEXISTS
If an object with the given ID already exists, the existing object should be replaced.

CO_UPDATEIFEXISTS
If an object with the given ID already exists, the existing object should be updated with the new information.

rc (HOBJECT) - returns
Handle to the created object.

NULLHANDLE
Error occurred.
Other
A handle to the object created. This handle is persistent and can be used for the WinSetObjectData and WinDestroyObject function calls.

WinCreateObject - Remarks

The *pszSetupString* contains a series of "keyname=value" pairs, that change the behavior of the object. "keynames" are separated by semicolons, and "values" are separated by commas.

```
"key=value;key2=value1,value2;"
```

If you want a literal comma or a literal semicolon inside one of your fields you must type the following:

^,	A literal comma.
^;	A literal semicolon.

Each object class documents the keynames and the parameters it expects to see immediately following. See the following keyname-value pairs tables:

- WPColorPalette
- WPDisk
- WPFolder
- WPFontPalette
- WPKeyboard
- WPLaunchPad
- WPPalette
- WPPrinter

- WPRPrinter
- WPProgram
- WPProgramFile
- WPSchemePalette
- WPShadow
- WPObject

Note that ALL parameters have safe defaults, so it is never necessary to pass unnecessary parameters to an object.

For more information about object classes, see the *Workplace Shell Programming Guide* .

Using HOBJECT for .INI files or files in which an application uses a rename/save/delete sequence is not supported.

WinCreateObject - Related Functions

Related Functions

- WinDeregisterObjectClass
- WinDestroyObject
- WinRegisterObjectClass
- WinReplaceObjectClass
- WinSetObjectData

WinCreateObject - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Related Functions
Glossary

WinCreateShadow

WinCreateShadow - Syntax

This function is specific to OS/2 Version 3.0 or higher.

This function creates a shadow of an object and places it in a specified location.

```
#define INCL_WINWORKPLACE
#include <os2.h>

HOBJECT    hObjectofObject;  /*  Handle of the object from which the shadow is to be created. */
HOBJECT    hObjectofDest;    /*  Handle of the folder into which hObjectofObject is to be placed. */
ULONG      ulReserved;       /*  Reserved value; must be 0. */
HOBJECT    rc;               /*  Handle of the newly created shadow object. */

rc = WinCreateShadow(hObjectofObject, hObjectofDest,
       ulReserved);
```

WinCreateShadow Parameter - hObjectofObject

hObjectofObject (HOBJECT) - input
Handle of the object from which the shadow is to be created.

WinCreateShadow Parameter - hObjectofDest

hObjectofDest (HOBJECT) - input
Handle of the folder into which *hObjectofObject* is to be placed.

WinCreateShadow Parameter - ulReserved

ulReserved (ULONG) - input
Reserved value; must be 0.

WinCreateShadow Return Value - rc

rc (HOBJECT) - returns
Handle of the newly created shadow object.

A return value of NULLHANDLE indicates that either *hObjectofDest* is NULLHANDLE or an object with the same name as *hObjectofObject* exists in the destination folder.

WinCreateShadow - Parameters

hObjectofObject (HOBJECT) - input
Handle of the object from which the shadow is to be created.

hObjectofDest (HOBJECT) - input
Handle of the folder into which *hObjectofObject* is to be placed.

ulReserved (ULONG) - input
Reserved value; must be 0.

rc (HOBJECT) - returns
Handle of the newly created shadow object.

A return value of NULLHANDLE indicates that either *hObjectofDest* is NULLHANDLE or an object with the same name as

hObjectofObject exists in the destination folder.

WinCreateShadow - Remarks

Using HOBJECT for .INI files or files in which an application uses a rename/save/delete sequence is not supported. Its REXX counterpart is SysCreateShadowObject.

WinCreateShadow - Related Functions

Related Functions

- WinCreateObject
- WinDestroyObject
- WinMoveObject
- WinQueryObjectWindow
- WinSaveObject

WinCreateShadow - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Related Functions
Glossary

WinDeregisterObjectClass

WinDeregisterObjectClass - Syntax

This function deregisters (removes) a workplace object class.

```
#define INCL_WINWORKPLACE
#include <os2.h>

PSZ     pszClassName;  /*  Pointer to class name. */
BOOL    rc;            /*  Success indicator. */

rc = WinDeregisterObjectClass(pszClassName);
```

WinDeregisterObjectClass Parameter - pszClassName

pszClassName (PSZ) - input
Pointer to class name.

A pointer to a zero-terminated string which contains the name of the object class being removed from the workplace.

WinDeregisterObjectClass Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinDeregisterObjectClass - Parameters

pszClassName (PSZ) - input
Pointer to class name.

A pointer to a zero-terminated string which contains the name of the object class being removed from the workplace.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinDeregisterObjectClass - Remarks

Workplace object classes are not deleted unless the application issues a WinDeregisterObjectClass. Object classes will be automatically registered when a dynamic-link library containing an object definition is added to the system. The only advantage of deregistering an object class is to optimize the system performance. All registered classes are maintained in the OS2.INI and are cached upon system initialization. If the class is no longer needed, it should be removed.

WinDeregisterObjectClass - Related Functions

Related Functions

- WinCreateObject
- WinRegisterObjectClass
- WinReplaceObjectClass

WinDeregisterObjectClass - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Related Functions
Glossary

WinDestroyObject

WinDestroyObject - Syntax

This function is called to delete a workplace object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

HOBJECT    object;  /*  Handle to a workplace object. */
BOOL       rc;      /*  Success indicator. */

rc = WinDestroyObject(object);
```

WinDestroyObject Parameter - object

object (HOBJECT) - input
Handle to a workplace object.

WinDestroyObject Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.

FALSE
Error occurred.

WinDestroyObject - Parameters

object (HOBJECT) - input
Handle to a workplace object.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

WinDestroyObject - Remarks

The WinDestroyObject function will permanently remove an object that was created with the WinCreateObject function.

Using HOBJECT for .INI files or files in which an application uses a rename/save/delete sequence is not supported.

WinDestroyObject - Related Functions

Related Functions

- WinCreateObject
- WinSetObjectData
- WinDestroyObject

WinDestroyObject - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Related Functions
Glossary

WinEnumObjectClasses

WinEnumObjectClasses - Syntax

The WinEnumObjectClasses function will return a list of all workplace object classes that have been registered.

```
#define INCL_WINWORKPLACE
#include <os2.h>

POBJCLASS    pObjClass;  /*  Pointer to object class. */
PULONG       pSize;      /*  Length of the pObjClass buffer in bytes. */
BOOL         rc;         /*  Success indicator. */

rc = WinEnumObjectClasses(pObjClass, pSize);
```

WinEnumObjectClasses Parameter - pObjClass

pObjClass (POBJCLASS) - input
Pointer to object class.

A pointer to a buffer to be filled with information about the registered workplace object classes.

WinEnumObjectClasses Parameter - pSize

pSize (PULONG) - in/out
Length of the *pObjClass* buffer in bytes.

If *pObjClass* is NULL, the actual size of *pObjClass* is returned in *pSize*

WinEnumObjectClasses Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinEnumObjectClasses - Parameters

pObjClass (POBJCLASS) - input
Pointer to object class.

A pointer to a buffer to be filled with information about the registered workplace object classes.

pSize (PULONG) - in/out
Length of the *pObjClass* buffer in bytes.

If *pObjClass* is NULL, the actual size of *pObjClass* is returned in *pSize*

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinEnumObjectClasses - Remarks

WinEnumObjectClasses will return a buffer containing all workplace object classes that are currently registered with the system. Workplace object classes are registered with the system through the function call WinRegisterObjectClass.

WinEnumObjectClasses - Related Functions

Related Functions

- WinRegisterObjectClass
- WinReplaceObjectClass

WinEnumObjectClasses - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Related Functions
Glossary

WinFreeFileIcon

WinFreeFileIcon - Syntax

This function frees an icon pointer that was originally allocated by WinLoadFileIcon.

```
#define INCL_WINWORKPLACE
#include <os2.h>

HPOINTER    hptr;  /*  A pointer to an icon loaded by WinLoadFileIcon. */
BOOL        rc;    /*  Success indicator. */
```

```
rc = WinFreeFileIcon(hptr);
```

WinFreeFileIcon Parameter - hptr

hptr (HPOINTER) - input
A pointer to an icon loaded by WinLoadFileIcon.

WinFreeFileIcon Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinFreeFileIcon - Parameters

hptr (HPOINTER) - input
A pointer to an icon loaded by WinLoadFileIcon.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinFreeFileIcon - Related Functions

Related Functions

- WinSetFileIcon
- WinLoadFileIcon

WinFreeFileIcon - Topics

Select an item:
Syntax

Parameters
Returns
Related Functions
Glossary

WinIsSOMDDReady

WinIsSOMDDReady - Syntax

This function returns the state of the DSOM daemon (SOMDD), started by the Workplace Shell process using WinRestartSOMDD.

```
#define INCL_WPCLASS
#include <os2.h>

BOOL    fReady;  /*  SOMDD-state indicator. */

fReady = WinIsSOMDDReady();
```

WinIsSOMDDReady Return Value - fReady

fReady (BOOL) - returns
SOMDD-state indicator.

TRUE
SOMDD has been started by the Workplace Shell process.

FALSE
SOMDD has not been started by the Workplace Shell process.

WinIsSOMDDReady - Parameters

fReady (BOOL) - returns
SOMDD-state indicator.

TRUE
SOMDD has been started by the Workplace Shell process.

FALSE
SOMDD has not been started by the Workplace Shell process.

WinIsSOMDDReady - Remarks

This function returns the state of the DSOM daemon started only by the Workplace Shell process using a call to WinRestartSOMDD. This

does not include the status of the DSOM daemon if started by any other process.

Note: This function requires that the Presentation Manager Shell is up and running.

WinIsSOMDDReady - Related Functions

Related Functions

- WinIsWPDServerReady
- WinRestartSOMDD
- WinRestartWPDServer

WinIsSOMDDReady - Example Code

This example starts the DSOM daemon and, a short time later, checks to see if it indeed has started successfully.

```
#define  INCL_WPCLASS
#include <os2.h>

enum {OFF, ON};

WinRestartSOMDD(ON);
   .
   .
   .
if ( WinIsSOMDDReady() )
   somPrintf ("SOMDD is running\n")
else
   somPrintf ("SOMDD failed to start, possiblly started already"
               "by another process\n");
```

WinIsSOMDDReady - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Example Code
Related Functions
Glossary

WinIsWPDServerReady

WinIsWPDServerReady - Syntax

This function returns the state of the Workplace Shell DSOM server.

```
#define INCL_WPCLASS
#include <os2.h>

BOOL    fReady;  /*  DSOM server-state indicator. DSOM Server status. */

fReady = WinIsWPDServerReady();
```

WinIsWPDServerReady Return Value - fReady

fReady (BOOL) - returns
DSOM server-state indicator. DSOM Server status.

TRUE
Workplace Shell DSOM Server is ready.
FALSE
Workplace Shell DSOM Server is not ready.

WinIsWPDServerReady - Parameters

fReady (BOOL) - returns
DSOM server-state indicator. DSOM Server status.

TRUE
Workplace Shell DSOM Server is ready.
FALSE
Workplace Shell DSOM Server is not ready.

WinIsWPDServerReady - Remarks

This function returns the ready status of the Workplace Shell DSOM Server.

Note: This function requires that the PM Shell is up and running.

WinIsWPDServerReady - Related Functions

Related Functions

- WinIsSOMDDReady
- WinRestartSOMDD
- WinRestartWPDServer

WinIsWPDServerReady - Example Code

This example stops the Workplace Shell DSOM Server, then waits until the server has terminate before stopping the DSOM daemon.

```
#define  INCL_WPCLASS
#include <os2.h>

ULONG  count=0;
enum   {OFF, ON};

WinRestartWPDServer(OFF);

/* Make sure the server thread has terminated completely before */
/* bring down the DSOM daemon                                   */
while ( WinIsWPDServerReady() )
{
   HEV hev;

   /* First create a private, reset, event semaphore.           */
   DosCreateEventSem( (PSZ)NULL, &hev, 0, FALSE);

   /* Wait for 1 second; then try again for a max. of 30 sec.   */
   DosWaitEventSem (hev, 1000);
   if (count++ > 30)
      break;
}
WinRestartSOMDD (OFF);
```

WinIsWPDServerReady - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Example Code
Related Functions
Glossary

WinLoadFileIcon

WinLoadFileIcon - Syntax

This function returns a pointer to an icon which is associated with the file specified by *pszFileName* .

```
#define INCL_WINWORKPLACE
#include <os2.h>

PSZ         pszFileName;  /*  Pointer to file name. */
BOOL        fPrivate;     /*  Icon usage flag. */
HPOINTER    hIcon;        /*  Icon handle. */
```

```
hIcon = WinLoadFileIcon(pszFileName, fPrivate);
```

WinLoadFileIcon Parameter - pszFileName

pszFileName (PSZ) - input
Pointer to file name.

A pointer to a zero-terminated string which contains the name of the file whose icon will be loaded.

WinLoadFileIcon Parameter - fPrivate

fPrivate (BOOL) - input
Icon usage flag.

TRUE
A private copy of this icon is requested. This flag should be used if the application needs to modify the icon.

FALSE
A shared pointer to this icon is requested. This flag should be used if application needs to display the icon without modifying it. This should be used whenever possible to optimize system resource use.

WinLoadFileIcon Return Value - hIcon

hIcon (HPOINTER) - returns
Icon handle.

NULL
Error occurred.

OTHER
Handle to an icon.

WinLoadFileIcon - Parameters

pszFileName (PSZ) - input
Pointer to file name.

A pointer to a zero-terminated string which contains the name of the file whose icon will be loaded.

fPrivate (BOOL) - input
Icon usage flag.

TRUE
A private copy of this icon is requested. This flag should be used if the application needs to modify the icon.

FALSE
A shared pointer to this icon is requested. This flag should be used if application needs to display the icon without

modifying it. This should be used whenever possible to optimize system resource use.

hIcon (HPOINTER) - returns
Icon handle.

NULL
Error occurred.
OTHER
Handle to an icon.

WinLoadFileIcon - Remarks

The icon will be retrieved in the following order until an icon has been found:

- .ICON extended attribute
- .ICO file in same directory with same prefix
- Application specific icon (if PM executable or MS Windows* executable)
- PM application icon (if PM executable)
- MS Windows* application icon (if MS Windows* application executable)
- OS/2 application icon (if OS/2 full-screen only executable)
- OS/2 window icon (if OS/2 window compatible executable)
- DOS windowed application icon (if DOS windowed executable)
- Program application (if unknown type executable)
- Data icon specified by associated application
- Data icon of associated application
- Data file icon (if not program or directory)
- Directory icon (if directory)

The HPOINTER returned in *fPrivate* should be freed by the caller via WinFreeFileIcon when it is no longer being used.

WinLoadFileIcon - Related Functions

Related Functions

- WinSetFileIcon
- WinFreeFileIcon

WinLoadFileIcon - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Related Functions
Glossary

WinMoveObject

WinMoveObject - Syntax

This function is specific to OS/2 Version 3.0 or higher.

This function moves an object from its existing location to a specified new destination.

```
#define INCL_WINWORKPLACE
#include <os2.h>

HOBJECT    hObjectofObject;  /*  Handle of the Workplace Shell object being moved. */
HOBJECT    hObjectofDest;    /*  Handle of the destination folder into which hObjectofObject is to be moved. */
ULONG      ulReserved;       /*  Reserved value; must be 0. */
HOBJECT    rc;               /*  Handle of the source object being moved. */

rc = WinMoveObject(hObjectofObject, hObjectofDest,
       ulReserved);
```

WinMoveObject Parameter - hObjectofObject

hObjectofObject (HOBJECT) - input
Handle of the Workplace Shell object being moved.

WinMoveObject Parameter - hObjectofDest

hObjectofDest (HOBJECT) - input
Handle of the destination folder into which *hObjectofObject* is to be moved.

WinMoveObject Parameter - ulReserved

ulReserved (ULONG) - input
Reserved value; must be 0.

WinMoveObject Return Value - rc

rc (HOBJECT) - returns
Handle of the source object being moved.

A return value of NULLHANDLE indicates that either *hObjectofDest* is NULLHANDLE or an object with the same name as *hObjectofObject* exists in the destination folder.

WinMoveObject - Parameters

hObjectofObject (HOBJECT) - input
Handle of the Workplace Shell object being moved.

hObjectofDest (HOBJECT) - input
Handle of the destination folder into which *hObjectofObject* is to be moved.

ulReserved (ULONG) - input
Reserved value; must be 0.

rc (HOBJECT) - returns
Handle of the source object being moved.

A return value of NULLHANDLE indicates that either *hObjectofDest* is NULLHANDLE or an object with the same name as *hObjectofObject* exists in the destination folder.

WinMoveObject - Remarks

Using HOBJECT for .INI files or files in which an application uses a rename/save/delete sequence is not supported. Its REXX counterpart is SysMoveObject.

WinMoveObject - Errors

Possible returns from WinGetLastError

WPERR_INVALID_FLAGS (0x1719)
An invalid flag was specified.

WinMoveObject - Related Functions

Related Functions

- WinCopyObject
- WinCreateObject
- WinDestroyObject
- WinQueryObjectWindow
- WinSaveObject

WinMoveObject - Example Code

This example moves the drives object onto the Desktop.

```
#define INCL_WINWORKPLACE
#include "os2.h"
```

```
HOBJECT    hObjectofDest;
HOBJECT    hObjectofObject;
HOBJECT    hObjectofResult;

hObjectofObject = WinQueryObject("<WP_DRIVES>");
if (hObjectofObject != NULL)
{
  /* WinQueryObject of Drives was successful */
  hObjectofDest = WinQueryObject("<WP_DESKTOP>");

  if (hObjectofDest != NULL)
  {
    /* WinQueryObject of Startup was successful */
    hObjectofResult = WinMoveObject(hObjectofObject, hObjectofDest, 0);

    if (hObjectofResult != NULL)
    {
      /* Drives Object was successfully copied to the Desktop */
    }
    else
    {
      /* Move failed */
    }
  }
}
```

WinMoveObject - Topics

Select an item:

Syntax
Parameters
Returns
Errors
Remarks
Example Code
Related Functions
Glossary

WinNotebookButtonFromID

WinNotebookButtonFromID - Syntax

This function is specific to Version 4, or higher, of the OS/2 operating system.

This function exports a convience function to find BS_NOTEBOOKBUTTON buttons.

```
#define INCL_WINWORKPLACE
#include <os2.h>

HWND     hwndDlg;  /*  Handle of the dialog window of the properties page. */
ULONG    id;       /*  ID of the button on the properties page. page. */
HWND     rc;       /*  Return Code */

rc = WinNotebookButtonFromID(hwndDlg, id);
```

WinNotebookButtonFromID Parameter - hwndDlg

hwndDlg (HWND) - input
Handle of the dialog window of the properties page.

WinNotebookButtonFromID Parameter - id

id (ULONG) - input
ID of the button on the properties page. page.

WinNotebookButtonFromID Return Value - rc

rc (HWND) - returns
Return Code

TRUE	Successful completion.
FALSE	Error occurred.

WinNotebookButtonFromID - Parameters

hwndDlg (HWND) - input
Handle of the dialog window of the properties page.

id (ULONG) - input
ID of the button on the properties page. page.

rc (HWND) - returns
Return Code

TRUE	Successful completion.
FALSE	Error occurred.

WinNotebookButtonFromID - Usage

Given the HWND of the dialog of the properties page, the ID of the button, and the button of the properties page (for example, Undo, Default, or Help), WinNotebookButtonFromID, returns the HWND for that button. This is used with the buttons of the BS_NOTEBOOKBUTTON style. Ownership of buttons of this style is changed from the dialog to the notebook after the dialog is loaded.

WinNotebookButtonFromID - Related Functions

Related Functions

- WinWaitForShell

WinNotebookButtonFromID - Topics

Select an item:
Syntax
Parameters
Returns
Usage
Related Functions
Glossary

WinOpenObject

WinOpenObject - Syntax

This function is specific to OS/2 Version 3.0 or higher.

This function eithers opens a view of the given object or surfaces an existing view.

```
#define INCL_WINWORKPLACE
#include <os2.h>

HOBJECT    hObjectofObject;  /*  Handle to a Workplace Shell object to be opened. */
ULONG      ulView;           /*  View in which to open the specified object. */
BOOL       fFlags;           /*  Flags. */
BOOL       rc;               /*  Success indicator. */

rc = WinOpenObject(hObjectofObject, ulView,
       fFlags);
```

WinOpenObject Parameter - hObjectofObject

hObjectofObject (HOBJECT) - input
Handle to a Workplace Shell object to be opened.

WinOpenObject Parameter - ulView

ulView (ULONG) - input
View in which to open the specified object.

OPEN_SETTINGS

OPEN_TREE

OPEN_DEFAULT

OPEN_CONTENTS

OPEN_DETAILS

WinOpenObject Parameter - fFlags

fFlags (BOOL) - input
Flags.

TRUE
Open a view of the object which already exists and supports concurrent views, by calling wpViewObject.

FALSE
Open a view of this object by calling wpOpen.

WinOpenObject Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion

FALSE
Error occurred.

WinOpenObject - Parameters

hObjectofObject (HOBJECT) - input
Handle to a Workplace Shell object to be opened.

ulView (ULONG) - input
View in which to open the specified object.

OPEN_SETTINGS

OPEN_TREE

OPEN_DEFAULT

OPEN_CONTENTS

OPEN_DETAILS

fFlags (BOOL) - input
Flags.

TRUE
Open a view of the object which already exists and supports concurrent views, by calling wpViewObject.

FALSE
Open a view of this object by calling wpOpen.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion

FALSE
Error occurred.

WinOpenObject - Remarks

Using HOBJECT for .INI files or files in which an application uses a rename/save/delete sequence is not supported. Its REXX counterpart is SysOpenObject. If concurrent views is off, wpViewObject will resurface an existing object.

WinOpenObject - Related Functions

Related Functions

- WinCreateObject
- WinDestroyObject
- WinSaveObject

WinOpenObject - Example Code

```
#define INCL_WINWORKPLACE
#include "os2.h"

HOBJECT hObject;
ULONG   ulView;
BOOL    fFlags
BOOL    fSuccess;

hObject = WinQueryObject("<WP_DESKTOP>");
if (hObject != NULL)
  {
    /* WinQueryObject was successful */

    fSuccess = WinOpenObject(hObject, OPEN_SETTINGS,TRUE);

    if (fSuccess)
    {
       /* If concurrent views is off and a settings view of the desktop already
        * exits, then it was successfully resurfaced.
        *
        * If concurrent views is on then another settings view of the desktop
        * was successfully opened
       */
    }
    else
    {
       /* WinOpenObject failed */
```

```
}
```

WinOpenObject - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Example Code
Related Functions
Glossary

WinQueryActiveDesktopPathname

WinQueryActiveDesktopPathname - Syntax

This function is specific to OS/2 Version 3.0 or higher.

This function returns the directory specification of the active desktop.

```
#define INCL_WINWORKPLACE
#include <os2.h>

PSZ      pszPathName;  /*  Memory allocated by caller in which directory specification is written. */
ULONG    ulSize;       /*  Number of bytes pointed to by pszPathName. */
BOOL     rc;           /*  Success indicator. */

rc = WinQueryActiveDesktopPathname(pszPathName,
       ulSize);
```

WinQueryActiveDesktopPathname Parameter - pszPathName

pszPathName (PSZ) - output
Memory allocated by caller in which directory specification is written.

WinQueryActiveDesktopPathname Parameter - ulSize

ulSize (ULONG) - input

Number of bytes pointed to by *pszPathName* .

WinQueryActiveDesktopPathname Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinQueryActiveDesktopPathname - Parameters

pszPathName (PSZ) - output
Memory allocated by caller in which directory specification is written.

ulSize (ULONG) - input
Number of bytes pointed to by *pszPathName* .

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinQueryActiveDesktopPathname - Remarks

This function is used to find the directory specification of the current desktop. The current desktop is not always \DESKTOP of the boot drive.

WinQueryActiveDesktopPathname - Errors

Possible returns from WinGetLastError

PMERR_INVALID_PARAMETER (0x1645)
An application parameter value is invalid for its converted PM type. For example: a 4-byte value outside the range -32,768 to +32,767 cannot be converted to a SHORT, and a negative number cannot be converted to a ULONG or USHORT.

WinQueryActiveDesktopPathname - Example Code

This example finds the directory specification of the current desktop.

```
#define INCL_WINWORKPLACE
#include <os2.h>

CHAR    szPath[CCHMAXPATH + 1];
BOOL    fSuccess;

fSuccess = WinQueryActiveDesktopPathname(szPath,sizeof(szPath));
if (fSuccess)
{
  /* WinQueryActiveDesktopPathname was successful */
}
else
{
  /* WinQueryActiveDesktopPathname failed */
}
```

WinQueryActiveDesktopPathname - Topics

Select an item:

Syntax
Parameters
Returns
Errors
Remarks
Example Code
Glossary

WinQueryObject

WinQueryObject - Syntax

The WinQueryObject function returns a handle to the given object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

PSZ        pObjectID;  /*  The object ID of an existing object. */
HOBJECT    hObject;    /*  MRESULT. */

hObject = WinQueryObject(pObjectID);
```

WinQueryObject Parameter - pObjectID

pObjectID (PSZ) - input
The object ID of an existing object.

The object ID of an existing object, for example <LOCATION_DESKTOP>, or alternatively the fully qualified filename of any file or directory.

WinQueryObject Return Value - hObject

hObject (HOBJECT) - returns
MRESULT.

Persistent object handle, or NULLHANDLE if the object does not exist or could not be awakened.

WinQueryObject - Parameters

pObjectID (PSZ) - input
The object ID of an existing object.

The object ID of an existing object, for example <LOCATION_DESKTOP>, or alternatively the fully qualified filename of any file or directory.

hObject (HOBJECT) - returns
MRESULT.

Persistent object handle, or NULLHANDLE if the object does not exist or could not be awakened.

WinQueryObject - Remarks

This function allows you to obtain the persistant object handle for any file object, by passing the fully qualified filename. Similarly any objects' handle can be retrieved if its object ID string is passed. Once a program has an object handle, it is able to change the objects state by using the WinSetObjectData function or delete the object using the WinDestroyObject function. Note that valid object ID strings must always start with the "<" character and be terminated by the ">" character, and are thus invalid file system names.

Using HOBJECT for .INI files or files in which an application uses a rename/save/delete sequence is not supported.

WinQueryObject - Related Functions

Related Functions

- WinCreateObject
- WinDestroyObject
- WinSetObjectData

WinQueryObject - Topics

Select an item:
Syntax
Parameters
Returns

Remarks
Related Functions
Glossary

WinQueryObjectPath

WinQueryObjectPath - Syntax

This function is specific to OS/2 Version 3.0 or higher.

This function returns the directory specification of a given object handle.

```
#define INCL_WINWORKPLACE
#include <os2.h>

HOBJECT    hObject;      /*  Object handle of the object whose file/directory specification is to be returned. *
PSZ        pszPathname;  /*  Memory allocated by caller in which directory specification is written. */
ULONG      ulSize;       /*  Number of bytes pointed to by pszPathname. */
BOOL       rc;           /*  Success indicator. */

rc = WinQueryObjectPath(hObject, pszPathname,
       ulSize);
```

WinQueryObjectPath Parameter - hObject

hObject (HOBJECT) - input
Object handle of the object whose file/directory specification is to be returned.

WinQueryObjectPath Parameter - pszPathname

pszPathname (PSZ) - output
Memory allocated by caller in which directory specification is written.

WinQueryObjectPath Parameter - ulSize

ulSize (ULONG) - input
Number of bytes pointed to by *pszPathname* .

WinQueryObjectPath Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinQueryObjectPath - Parameters

hObject (HOBJECT) - input
Object handle of the object whose file/directory specification is to be returned.

pszPathname (PSZ) - output
Memory allocated by caller in which directory specification is written.

ulSize (ULONG) - input
Number of bytes pointed to by *pszPathname* .

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinQueryObjectPath - Remarks

This function is used to find the file/directory specification of a given object handle.

Using HOBJECT for .INI files or files in which an application uses a rename/save/delete sequence is not supported.

WinQueryObjectPath - Errors

Possible returns from WinGetLastError

PMERR_INVALID_PARAMETERS (0x1208)
An application parameter value is invalid for its converted PM type. For example: a 4-byte value outside the range -32 768 to +32 767 cannot be converted to a SHORT, and a negative number cannot be converted to a ULONG or USHORT.

WinQueryObjectPath - Example Code

This example finds the file/directory of the <WP_OS2SYS> object..

```
#define INCL_WINWORKPLACE
#include <os2.h>

HOBJECT hObject;
CHAR    szPath[CCHMAXPATH + 1];
BOOL    fSuccess;

hObject = WinQueryObject("<WP_OS2SYS>");
if (hObject != NULL)
{
  /* WinQueryObject was successful */

  fSuccess = WinQueryObjectPath(hObject,szPath,sizeof(szPath));
  if (fSuccess)
  {
    /* WinQueryObjectPath was successful */
  }
  else
  {
    /* WinQueryObjectPath failed */
  }
}
else
{
  /* WinQueryObject failed */
}
```

WinQueryObjectPath - Topics

Select an item:

Syntax
Parameters
Returns
Errors
Remarks
Example Code
Glossary

WinRegisterObjectClass

WinRegisterObjectClass - Syntax

The WinRegisterObjectClass function registers a workplace object class.

```
#define INCL_WINWORKPLACE
#include <os2.h>

PSZ     pClassName;  /*  A pointer to object class being registered. */
PSZ     pModname;    /*  A pointer to DLL name. */
BOOL    rc;          /*  Success indicator. */

rc = WinRegisterObjectClass(pClassName, pModname);
```

WinRegisterObjectClass Parameter - pClassName

pClassName (PSZ) - input
A pointer to object class being registered.

A pointer to a zero-terminated string which contains the name of the object class being registered in the workplace.

WinRegisterObjectClass Parameter - pModname

pModname (PSZ) - input
A pointer to DLL name.

A pointer to a zero-terminated string which contains the name of the DLL which holds the object definition.

WinRegisterObjectClass Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinRegisterObjectClass - Parameters

pClassName (PSZ) - input
A pointer to object class being registered.

A pointer to a zero-terminated string which contains the name of the object class being registered in the workplace.

pModname (PSZ) - input
A pointer to DLL name.

A pointer to a zero-terminated string which contains the name of the DLL which holds the object definition.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinRegisterObjectClass - Remarks

The DLL must be one created using the IBM System Object Model. Object classes will automatically be added to the system when installing a DLL which contains an object definition. Generally, it is not required for the object DLL to be present at the time WinRegisterObjectClass is called. However, if the object class overrides wpclsQueryInstanceType or wpclsQueryInstanceFilter, the DLL must be present at the time of the class registration.

WinRegisterObjectClass - Related Functions

Related Functions

- WinCreateObject
- WinDeregisterObjectClass
- WinReplaceObjectClass

WinRegisterObjectClass - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Related Functions
Glossary

WinReplaceObjectClass

WinReplaceObjectClass - Syntax

The WinReplaceObjectClass function replaces a registered class with another registered class. If *fReplace* is FALSE, *pOldClassName* will revert back to its original definition.

```
#define INCL_WINWORKPLACE
#include <os2.h>

PSZ     pOldClassName;  /*  Pointer to class name. */
PSZ     pNewClassName;  /*  Pointer to new class name. */
BOOL    fReplace;       /*  Function replacement flag. */
BOOL    rc;             /*  Success indicator. */

rc = WinReplaceObjectClass(pOldClassName,
       pNewClassName, fReplace);
```

WinReplaceObjectClass Parameter - pOldClassName

pOldClassName (PSZ) - input
Pointer to class name.

A pointer to a zero-terminated string which contains the name of the object class being replaced by *pNewClassName* in the workplace.

WinReplaceObjectClass Parameter - pNewClassName

pNewClassName (PSZ) - input
Pointer to new class name.

A pointer to a zero-terminated string which contains the name of the object class replacing the *pOldClassName* class.

WinReplaceObjectClass Parameter - fReplace

fReplace (BOOL) - input
Function replacement flag.

TRUE
Replace the function of class *pOldClassName* with the function of the class *pNewClassName* .

FALSE
Undo the replacement of the *pOldClassName* with *pNewClassName* by restoring the *pOldClassName* back to its original functionality.

WinReplaceObjectClass Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinReplaceObjectClass - Parameters

pOldClassName (PSZ) - input
Pointer to class name.

A pointer to a zero-terminated string which contains the name of the object class being replaced by *pNewClassName* in the workplace.

pNewClassName (PSZ) - input
Pointer to new class name.

A pointer to a zero-terminated string which contains the name of the object class replacing the *pOldClassName* class.

fReplace (BOOL) - input
Function replacement flag.

TRUE
Replace the function of class *pOldClassName* with the function of the class *pNewClassName* .

FALSE
Undo the replacement of the *pOldClassName* with *pNewClassName* by restoring the *pOldClassName* back to its original functionality.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion

FALSE
Error occurred.

WinReplaceObjectClass - Remarks

The class specified by *pNewClassName* must be a descendant of the class specified by *pOldClassName* , otherwise an error will be returned. Replacing an object is useful if it is desired to modify the behavior of objects which are instances of the class *pOldClassName* and which are not aware of the class *pNewClassName* .

WinReplaceObjectClass - Related Functions

Related Functions

- WinCreateObject
- WinDeregisterObjectClass
- WinRegisterObjectClass

WinReplaceObjectClass - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Related Functions
Glossary

WinRestartSOMDD

WinRestartSOMDD - Syntax

This function starts the DSOM daemon.

```
#define INCL_WPCLASS
#include <os2.h>

BOOL      fAction;  /*  Flag indicating action to perform on the DSOM daemon. */
APIRET    rc;       /*  Return code. */

rc = WinRestartSOMDD(fAction);
```

WinRestartSOMDD Parameter - fAction

fAction (BOOL) - input
Flag indicating action to perform on the DSOM daemon.

TRUE
Start the DSOM daemon.
FALSE
Stop the DSOM daemon.

WinRestartSOMDD Return Value - rc

rc (APIRET) - returns
Return code.

PMERR_OK (0x0000)
Successfully invoked.

PMERR_WPDSERVER_IS_ACTIVE (0x1056)
The Workplace Shell DSOM Server is active and running. It cannot stop the DSOM daemon until the Workplace Shell DSOM Server has stopped.

PMERR_SOMDD_IS_ACTIVE (0x1058)
The DSOM daemon has been started by the Workplace Shell process and is active and running; therefore, it cannot be started again.

PMERR_SOMDD_NOT_STARTED (0x1059)
The DSOM daemon failed to start. It might be active in another process.

WinRestartSOMDD - Parameters

fAction (BOOL) - input
Flag indicating action to perform on the DSOM daemon.

TRUE
Start the DSOM daemon.
FALSE
Stop the DSOM daemon.

rc (APIRET) - returns
Return code.

PMERR_OK (0x0000)
Successfully invoked.

PMERR_WPDSERVER_IS_ACTIVE (0x1056)
The Workplace Shell DSOM Server is active and running. It cannot stop the DSOM daemon until the Workplace Shell DSOM Server has stopped.

PMERR_SOMDD_IS_ACTIVE (0x1058)
The DSOM daemon has been started by the Workplace Shell process and is active and running; therefore, it cannot be started again.

PMERR_SOMDD_NOT_STARTED (0x1059)
The DSOM daemon failed to start. It might be active in another process.

WinRestartSOMDD - Remarks

This function starts the DSOM daemon in the Workplace Shell process as a background process invisible to the users. The status of the daemon can be determined at any time by calling WinIsSOMDDReady.

Note: This function requires that the PM Shell is up and running.

WinRestartSOMDD - Related Functions

Related Functions

- WinIsSOMDDReady
- WinIsWPDServerReady
- WinRestartWPDServer

WinRestartSOMDD - Example Code

This example starts the DSOM daemon within the Workplace Shell process.

```
#define  INCL_WPCLASS
#include <os2.h>

enum   {OFF, ON};
APIRET apiRtnCd;

apiRtnCd = WinRestartSOMDD(ON);
if (apiRtnCd == PMERR_SOMDD_NOT_STARTED)
   somPrintf ("DSOM Daemon failed to start; might be active already");
```

WinRestartSOMDD - Topics

Select an item:
Syntax
Parameters
Returns

Remarks
Example Code
Related Functions
Glossary

WinRestartWPDServer

WinRestartWPDServer - Syntax

This function starts the Workplace Shell DSOM server.

```
#define INCL_WPCLASS
#include <os2.h>

BOOL      fAction;  /*  Flag indicating the action to perform on the Workplace Shell DSOM server. */
APIRET    rc;       /*  Return code. */

rc = WinRestartWPDServer(fAction);
```

WinRestartWPDServer Parameter - fAction

fAction (BOOL) - input
Flag indicating the action to perform on the Workplace Shell DSOM server.

TRUE
Start the Workplace Shell DSOM server.

FALSE
Stop the Workplace Shell DSOM server.

WinRestartWPDServer Return Value - rc

rc (APIRET) - returns
Return code.

PMERR_OK (0x0000)
Successfully invoked.

PMERR_WPDSERVER_IS_ACTIVE (0x1056)
The Workplace Shell DSOM server is active and cannot be started again.

PMERR_WPDSERVER_NOT_STARTED (0x1057)
The Workplace Shell DSOM server could not be started, possibly related to a corrupted copy of WPDSRVP.DLL.

Other
Any other non-zero value could indicate that the WPDSRVP.DLL is corrupted or not found.

WinRestartWPDServer - Parameters

fAction (BOOL) - input
Flag indicating the action to perform on the Workplace Shell DSOM server.

TRUE
Start the Workplace Shell DSOM server.
FALSE
Stop the Workplace Shell DSOM server.

rc (APIRET) - returns
Return code.

PMERR_OK (0x0000)
Successfully invoked.

PMERR_WPDSERVER_IS_ACTIVE (0x1056)
The Workplace Shell DSOM server is active and cannot be started again.

PMERR_WPDSERVER_NOT_STARTED (0x1057)
The Workplace Shell DSOM server could not be started, possibly related to a corrupted copy of WPDSRVP.DLL.

Other
Any other non-zero value could indicate that the WPDSRVP.DLL is corrupted or not found.

WinRestartWPDServer - Remarks

This function requires that the DSOM daemon is started first and must be running successfully in order for the Workplace Shell DSOM server to become ready. The status of the Workplace Shell DSOM server can be obtained at any time using WinIsWPDServerReady. Once the DSOM server is ready, a client Workplace Shell DSOM server application can be executed.

Note: This function requires that the PM Shell is up and running.

WinRestartWPDServer - Related Functions

Related Functions

- WinIsSOMDDReady
- WinIsWPDServerReady
- WinRestartSOMDD

WinRestartWPDServer - Example Code

This example starts the DSOM daemon within the Workplace Shell process and then starts the Workplace Shell DSOM server.

```
#define  INCL_WINERRORS
#define  INCL_WPCLASS
#include <os2.h>

enum     {OFF, ON};
APIRET   apiRtnCd;
```

```
apiRtnCd=WinRestartSOMDD(ON);
if (apiRtnCd == PMERR_OK)
{
  apiRtnCd = WinRestartWPDServer(ON);
  if (apiRtnCd)
  {
    PERRINFO perriErrorInfo;

    if (apiRtnCd == PMERR_WPDSERVER_IS_ACTIVE)
    {
      somPrintf("The Workplace Shell DSOM server is already running");
      exit(2);
    }
    if (apiRtnCd == PMERR_WPDSERVER_NOT_STARTED)
    {
      somPrintf("Failed to start the Workplace Shell DSOM server");
      exit(2);
    }

    /* Obtain the error block                                    */
    /* WinGetErrorInfo is for DosLoadModule related errors only */
    perriErrorInfo = WinGetErrorInfo(hab);
    somPrintf("Loading WPDSRVP.DLL failed: %ld", perriErrorInfo->idError);
    WinFreeErrorInfo(perriErrorInfo);
    exit(99);
  }
  somPrintf("All's well!");
}
else
{
  somPrintf("Failed to start the DSOM daemon");
  exit(1);
}
```

WinRestartWPDServer - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Example Code
Related Functions
Glossary

WinRestoreWindowPos

WinRestoreWindowPos - Syntax

The WinRestoreWindowPos function will restore the size and position of the window specified by *hwnd* to the state it was in when WinStoreWindowPos was last called with the same *pAppName* and *pKeyName* .

```
#define INCL_WINWORKPLACE
#include <os2.h>

PSZ     pAppName;  /*  Pointer to application name. */
```

```
PSZ     pKeyName;  /*  Pointer to key name. */
HWND    hwnd;      /*  Window handle of the window to restore. */
BOOL    rc;        /*  Success indicator. */

rc = WinRestoreWindowPos(pAppName, pKeyName,
       hwnd);
```

WinRestoreWindowPos Parameter - pAppName

pAppName (PSZ) - input
Pointer to application name.

A pointer to a zero-terminated string which contains the application name.

WinRestoreWindowPos Parameter - pKeyName

pKeyName (PSZ) - input
Pointer to key name.

A pointer to a zero-terminated string which contains the key name.

WinRestoreWindowPos Parameter - hwnd

hwnd (HWND) - input
Window handle of the window to restore.

WinRestoreWindowPos Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinRestoreWindowPos - Parameters

pAppName (PSZ) - input
Pointer to application name.

A pointer to a zero-terminated string which contains the application name.

pKeyName (PSZ) - input
Pointer to key name.

A pointer to a zero-terminated string which contains the key name.

hwnd (HWND) - input
Window handle of the window to restore.

rc (BOOL) - returns
Success indicator.

TRUE	Successful completion
FALSE	Error occurred.

WinRestoreWindowPos - Remarks

This function will also restore presentation parameters which were saved by a previous call to WinStoreWindowPos.

WinRestoreWindowPos - Related Functions

Related Functions

- WinStoreWindowPos

WinRestoreWindowPos - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Related Functions
Glossary

WinSaveObject

WinSaveObject - Syntax

This function is specific to OS/2 Version 3.0 or higher.

This function saves the state of an object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

HOBJECT    hObject;  /*  Handle of the object to be saved. */
BOOL       fAsync;   /*  Asynchronous flag. */
BOOL       rc;       /*  Success indicator. */

rc = WinSaveObject(hObject, fAsync);
```

WinSaveObject Parameter - hObject

hObject (HOBJECT) - input
Handle of the object to be saved.

WinSaveObject Parameter - fAsync

fAsync (BOOL) - input
Asynchronous flag.

TRUE
Calls asynchronous save.

FALSE
Calls synchronous save.

WinSaveObject Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.

FALSE
Error occurred.

WinSaveObject - Parameters

hObject (HOBJECT) - input
Handle of the object to be saved.

fAsync (BOOL) - input
Asynchronous flag.

TRUE
Calls asynchronous save.

FALSE
Calls synchronous save.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

WinSaveObject - Remarks

If an asynchronous save is called, the object will be saved on a separate thread ("lazy written"); this is the preferred method for saving. Otherwise, the object is saved on the user interface thread.

Using HOBJECT for .INI files or files in which an application uses a rename/save/delete sequence is not supported.

WinSaveObject - Related Functions

Related Functions

- WinCopyObject
- WinCreateObject
- WinDestroyObject
- WinMoveObject
- WinQueryObjectWindow

WinSaveObject - Example Code

```
#define INCL_WINWORKPLACE
#include "os2.h"

HOBJECT  Object;
BOOL     fAsync = TRUE;
BOOL     fSuccess

hObject = WinQueryObject("<WP_DESKTOP>");
if (hObject != NULL)
{
  /* WinQueryObject was successful */
  fSuccess = WinSaveObject(hObject,fAsync);
  if (fSuccess)
  {
    /* The state of the Desktop was saved Asynchronously */
  }
  else
  {
    /* Asynchronous save of the Desktop failed */
  }
}
```

WinSaveObject - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Example Code
Related Functions
Glossary

WinSetFileIcon

WinSetFileIcon - Syntax

WinSetFileIcon sets the icon for the file specified by *pFileName* to the icon specified by *picon*.

```
#define INCL_WINPOINTERS
#include <os2.h>

PSZ          pFileName;  /*  Pointer to file name. */
PICONINFO    picon;      /*  A pointer to an ICONINFO structure containing an icon specification. */
BOOL         rc;         /*  Success indicator. */

rc = WinSetFileIcon(pFileName, picon);
```

WinSetFileIcon Parameter - pFileName

pFileName (PSZ) - input
Pointer to file name.

A pointer to a zero-terminated string which contains the name of the file whose icon will be set.

WinSetFileIcon Parameter - picon

picon (PICONINFO) - input
A pointer to an ICONINFO structure containing an icon specification.

WinSetFileIcon Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinSetFileIcon - Parameters

pFileName (PSZ) - input
Pointer to file name.

A pointer to a zero-terminated string which contains the name of the file whose icon will be set.

picon (PICONINFO) - input
A pointer to an ICONINFO structure containing an icon specification.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinSetFileIcon - Remarks

The specified icon is written to the file's .ICON extended attribute.

WinSetFileIcon - Related Functions

Related Functions

- WinLoadFileIcon
- WinFreeFileIcon

WinSetFileIcon - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Related Functions
Glossary

WinSetObjectData

WinSetObjectData - Syntax

The WinSetObjectData function is called to set data on a workplace object.

```
#define INCL_WINWORKPLACE
#include <os2.h>

HOBJECT    object;        /*  Handle to a workplace object. */
PSZ        pSetupString;  /*  Pointer to object-specific parameters. */
BOOL       rc;            /*  Success indicator. */

rc = WinSetObjectData(object, pSetupString);
```

WinSetObjectData Parameter - object

object (HOBJECT) - input
Handle to a workplace object.

WinSetObjectData Parameter - pSetupString

pSetupString (PSZ) - input
Pointer to object-specific parameters.

A pointer to a zero-terminated string which contains the object-specific parameters to the new object.

The *pSetupString* string is extracted when the wpSetup method is called. For a listing of setup strings, see individual object classes; See WPLaunchPad class definition for a discussion of setup string values used in Toolbar customization.

WinSetObjectData Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.

FALSE
Error occurred.

WinSetObjectData - Parameters

object (HOBJECT) - input
Handle to a workplace object.

pSetupString (PSZ) - input
Pointer to object-specific parameters.

A pointer to a zero-terminated string which contains the object-specific parameters to the new object.

The *pSetupString* string is extracted when the wpSetup method is called. For a listing of setup strings, see individual object classes; See WPLaunchPad class definition for a discussion of setup string values used in Toolbar customization.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion.
FALSE
Error occurred.

WinSetObjectData - Remarks

The WinSetObjectData function will change settings on an object that was created with the WinCreateObject function.

Using HOBJECT for .INI files or files in which an application uses a rename/save/delete sequence is not supported.

WinSetObjectData - Related Functions

Related Functions

- WinCreateObject
- WinSetObjectData
- WinDestroyObject

WinSetObjectData - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Related Functions
Glossary

WinShutdownSystem

WinShutdownSystem - Syntax

The WinShutdownSystem function will close down the system.

```
#define INCL_WINWORKPLACE
#include <os2.h>

HAB     hab;  /*  Anchor-block handle. */
HMQ     hmq;  /*  Message-queue handle. */
BOOL    rc;   /*  Success indicator. */

rc = WinShutdownSystem(hab, hmq);
```

WinShutdownSystem Parameter - hab

hab (HAB) - input
Anchor-block handle.

WinShutdownSystem Parameter - hmq

hmq (HMQ) - input
Message-queue handle.

WinShutdownSystem Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinShutdownSystem - Parameters

hab (HAB) - input
Anchor-block handle.

hmq (HMQ) - input
Message-queue handle.

rc (BOOL) - returns
Success indicator.

TRUE
: Successful completion

FALSE
: Error occurred.

WinShutdownSystem - Remarks

The WinShutdownSystem function will close all running applications and will then call DosShutdown.

Presentation Manager applications will receive a WM_SAVEAPPLICATION message prior to a WM_QUIT message.

When the system is restarted, all applications that were running when WinShutdownSystem was last called will be restarted.

WinShutdownSystem - Related Functions

Related Functions

- WinCancelShutdown

WinShutdownSystem - Example Code

This example performs an OS/2 System Shutdown from a program.

```
#define INCL_WINWORKPLACE
#define INCL_DOSFILEMGR
#define INCL_DOSERRORS
#include <os2.h>
#include <stdio.h>

int main(VOID)
  {

   HAB   hab      = NULLHANDLE;   /* Window handle */
   HMQ   hmq      = NULLHANDLE;   /* Message queue handle */
   BOOL  fSuccess = 0;            /* Win API success indicator */

   hab = WinInitialize( 0 );
   hmq = WinCreateMsgQueue( hab, 0 );

   /* Prevent our program from hanging the shutdown.  If this call is
      omitted, the system will wait for us to do a WinDestroyMsgQueue. */

   fSuccess = WinCancelShutdown( hmq, TRUE );

   /* Shutdown the system! */

   printf("System Shutdown will now be attempted...\n");
   fSuccess = WinShutdownSystem( hab, hmq );

   if (!fSuccess) {
      return 1;
   } else {
      return NO_ERROR;
   } /* endif */

   }
```

WinShutdownSystem - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Example Code
Related Functions
Glossary

WinStoreWindowPos

WinStoreWindowPos - Syntax

The WinStoreWindowPos function will save the current size and position of the window specified by *hwnd*.

```
#define INCL_WINWORKPLACE
#include <os2.h>

PSZ     pAppName;  /*  Pointer to application name. */
PSZ     pKeyName;  /*  Pointer to key name. */
HWND    hwnd;      /*  Window handle for the window to be stored. */
BOOL    rc;        /*  Success indicator. */

rc = WinStoreWindowPos(pAppName, pKeyName,
       hwnd);
```

WinStoreWindowPos Parameter - pAppName

pAppName (PSZ) - input
Pointer to application name.

A pointer to a zero-terminated string which contains the application name.

WinStoreWindowPos Parameter - pKeyName

pKeyName (PSZ) - input
Pointer to key name.

A pointer to a zero-terminated string which contains the key name.

WinStoreWindowPos Parameter - hwnd

hwnd (HWND) - input
Window handle for the window to be stored.

WinStoreWindowPos Return Value - rc

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinStoreWindowPos - Parameters

pAppName (PSZ) - input
Pointer to application name.

A pointer to a zero-terminated string which contains the application name.

pKeyName (PSZ) - input
Pointer to key name.

A pointer to a zero-terminated string which contains the key name.

hwnd (HWND) - input
Window handle for the window to be stored.

rc (BOOL) - returns
Success indicator.

TRUE
Successful completion
FALSE
Error occurred.

WinStoreWindowPos - Remarks

This function will also save the presentation parameters.

WinStoreWindowPos - Related Functions

Related Functions

- WinRestoreWindowPos

WinStoreWindowPos - Topics

Select an item:
Syntax
Parameters
Returns
Remarks
Related Functions
Glossary

WinWaitForShell

WinWaitForShell - Syntax

This function is specific to Version 4, or higher, of the OS/2 operating system.

This function allows applications to wait for the Shell to be available.

```
#define INCL_WINWORKPLACE
#include <os2.h>

ULONG    ulEvent;  /*  Event pointed to. */
BOOL     rc;       /*  Return Code */

rc = WinWaitForShell(ulEvent);
```

WinWaitForShell Parameter - ulEvent

ulEvent (ULONG) - input
Event pointed to.

WWFS_QUERY	0x80000000
WWFS_DESKTOPCREATED	1
WWFS_DESKTOPOPENED	2
WWFS_DESKTOPPOPULATED	3

WinWaitForShell Return Value - rc

rc (BOOL) - returns
Return Code

TRUE	Successful completion.
FALSE	Error occurred.

WinWaitForShell - Parameters

ulEvent (ULONG) - input
Event pointed to.

WWFS_QUERY	0x80000000
WWFS_DESKTOPCREATED	1
WWFS_DESKTOPOPENED	2
WWFS_DESKTOPPOPULATED	3

rc (BOOL) - returns
Return Code

TRUE	Successful completion.
FALSE	Error occurred.

WinWaitForShell - Usage

WinWaitForShell is used by an application when it needs to wait for the Workplace Shell to be available in order for the application to continue. The application can use WinWaitForShell to wait for the Workplace Shell to be ready to start receiving the create requests. By using the optional query bit, it can check on those conditions without waiting on them.

For example, the application might be started in STARTUP.CMD, and yet it wants to create objects on the Workplace Shell.

It can wait on one of three events:

- The desktop object has been created.
- The desktop object has been opened.
- The desktop object is fully populated.

WinWaitForShell - Related Functions

Related Functions

- WinNotebookButtonFromID

WinWaitForShell - Topics

Select an item:
Syntax
Parameters
Returns
Usage
Related Functions

Data Types

The following data types are used in Workplace Shell. They are listed in alphabetic order.

ACTIONS

Action button data structure.

```
typedef struct _ACTIONS {
  PSZ          pszTitle;  /*  String containing the title of the action button. */
  ULONG        ulMenuId;  /*  Desktop menu IDs. */
  HPOINTER     hIcon;     /*  The icon handle. */
} ACTIONS;

typedef ACTIONS *PACTIONS;
```

ACTIONS Field - pszTitle

pszTitle (PSZ)
String containing the title of the action button.

ACTIONS Field - ulMenuId

ulMenuId (ULONG)
Desktop menu IDs.

ACTIONS Field - hIcon

hIcon (HPOINTER)
The icon handle.

APIRET

Unsigned integer in the range 0 through 4 294 967 295.

```
typedef unsigned long APIRET;
```

BOOL

Boolean.

Valid values are:

- FALSE, which is 0
- TRUE, which is 1

```
typedef unsigned long BOOL;
```

BOOL32

Boolean.

Valid values are:

- FALSE, which is 0
- TRUE, which is 1

```
typedef unsigned long BOOL32;
```

BYTE

A byte.

```
typedef unsigned char BYTE;
```

CDATE

Structure that contains date information for a data element in the details view of a container control.

```
typedef struct _CDATE {
  UCHAR      day;    /*  Current day. */
  UCHAR      month;  /*  Current month. */
  USHORT     year;   /*  Current year. */
} CDATE;

typedef CDATE *PCDATE;
```

CDATE Field - day

day (UCHAR)
Current day.

CDATE Field - month

month (UCHAR)
Current month.

CDATE Field - year

year (USHORT)
Current year.

CELL

Class-specific cell data follows immediately afterward.

```
typedef struct _CELL {
  ULONG     cbData;  /*  Size of the data that follows. */
} CELL;

typedef CELL *PCELL;
```

CELL Field - cbData

cbData (ULONG)
Size of the data that follows.

Class-specific cell data follows immediately after a variable of this type. For example the font palette would store the ASCII name of the font, and the color palette would store the RGB color of the cell.

CHAR

Single-byte character.

```
#define CHAR char
```

CHARBUNDLE

Character-attributes bundle structure.

```
typedef struct _CHARBUNDLE {
  LONG       lColor;          /*  Character foreground color. */
  LONG       lBackColor;      /*  Character background color. */
  USHORT     usMixMode;       /*  Character foreground-mix mode. */
  USHORT     usBackMixMode;   /*  Character background-mix mode. */
  USHORT     usSet;           /*  Character set. */
  USHORT     usPrecision;     /*  Character precision. */
  SIZEF      sizfxCell;       /*  Character cell size. */
  POINTL     ptlAngle;        /*  Character angle. */
  POINTL     ptlShear;        /*  Character shear. */
  USHORT     usDirection;     /*  Character direction. */
  USHORT     usTextAlign;     /*  Text alignment. */
  FIXED      fxExtra;         /*  Character extra. */
  FIXED      fxBreakExtra;    /*  Character break extra. */
} CHARBUNDLE;

typedef CHARBUNDLE *PCHARBUNDLE;
```

CHARBUNDLE Field - lColor

lColor (LONG)
Character foreground color.

CHARBUNDLE Field - lBackColor

lBackColor (LONG)
Character background color.

CHARBUNDLE Field - usMixMode

usMixMode (USHORT)
Character foreground-mix mode.

CHARBUNDLE Field - usBackMixMode

usBackMixMode (USHORT)
Character background-mix mode.

CHARBUNDLE Field - usSet

usSet (USHORT)
Character set.

CHARBUNDLE Field - usPrecision

usPrecision (USHORT)
Character precision.

CHARBUNDLE Field - sizfxCell

sizfxCell (SIZEF)
Character cell size.

CHARBUNDLE Field - ptlAngle

ptlAngle (POINTL)
Character angle.

CHARBUNDLE Field - ptlShear

ptlShear (POINTL)
Character shear.

CHARBUNDLE Field - usDirection

usDirection (USHORT)

Character direction.

CHARBUNDLE Field - usTextAlign

usTextAlign (USHORT)
Text alignment.

CHARBUNDLE Field - fxExtra

fxExtra (FIXED)
Character extra.

CHARBUNDLE Field - fxBreakExtra

fxBreakExtra (FIXED)
Character break extra.

CLASS

An array of class objects. The final element of the array should be NULL.

```
typedef M_WPObject *CLASS;
```

CLASSCRITERIA

Points to a linked list of CRITERIA records for the given class.

```
typedef struct _CLASSCRITERIA {
  struct _CLASSCRITERIA      *pNext;         /*  Pointer to the next class criteria record. */
  struct _CLASSCRITERIA      *pPrev;         /*  Pointer to the previous class criteria record. */
  M_WPObject                 *Class;         /*  Pointer to the class object. */
  PCRITERIA                   pCriteria;     /*  Pointer to a linked list of CRITERIA records. */
  WPObject                   *NotifyObject;  /*  The owner of the criteria list. */
} CLASSCRITERIA;

typedef CLASSCRITERIA *PCLASSCRITERIA;
```

CLASSCRITERIA Field - pNext

pNext (struct _CLASSCRITERIA *)
Pointer to the next class criteria record.

CLASSCRITERIA Field - pPrev

pPrev (struct _CLASSCRITERIA *)
Pointer to the previous class criteria record.

CLASSCRITERIA Field - Class

Class (M_WPObject *)
Pointer to the class object.

CLASSCRITERIA Field - pCriteria

pCriteria (PCRITERIA)
Pointer to a linked list of CRITERIA records.

CLASSCRITERIA Field - NotifyObject

NotifyObject (WPObject *)
The owner of the criteria list.

CLASSDETAILS

Class details data structure.

```
typedef STRUCT _CLASSDETAILS {
  PSZ       pszAttribute;  /*  Translatable string. */
  PVOID     pSortRecord;   /*  Function pointer for the sort function for the attribute. */
} CLASSDETAILS;

typedef CLASSDETAILS *PCLASSDETAILS;
```

CLASSDETAILS Field - pszAttribute

pszAttribute (PSZ)
Translatable string.

Translatable string for a class attribute.

CLASSDETAILS Field - pSortRecord

pSortRecord (PVOID)
Function pointer for the sort function for the attribute.

CLASSFIELDINFO

Class field information data structure for use with the wpclsQueryDetailsInfo method.

```
typedef struct _CLASSFIELDINFO {
  ULONG                         cb;                 /*  Size of the CLASSFIELDINFO structure. */
  ULONG                         flData;             /*  Attributes of the field's data. */
  ULONG                         flTitle;            /*  Attributes of the field's title. */
  PVOID                         pTitleData;         /*  Title data. */
  ULONG                         ulReserved;         /*  Reserved. */
  PVOID                         pUserData;          /*  Pointer to user data. */
  struct _CLASSFIELDINFO       *pNextFieldInfo;     /*  Pointer to the next linked CLASSFIELDINFO structure. */
  ULONG                         cxWidth;            /*  Width of the field in pels. */
  ULONG                         offFieldData;       /*  Offset from beginning of this class's data for this fie
  ULONG                         ulLenFieldData;     /*  Width of data in bytes. */
  PFNOWNDRW                     pfnOwnerDraw;       /*  Ownerdraw procedure for Details View column. */
  ULONG                         flCompare;          /*  Flags. */
  PFNCOMPARE                    pfnCompare;         /*  Pointer to a comparison function. */
  ULONG                         DefaultComparison;  /*  Default comparison operator in the Include Page Criteri
  ULONG                         ulLenCompareValue;  /*  Maximum length of the compare data. */
  PVOID                         pDefCompareValue;   /*  The default value to be used for comparisons. */
  PVOID                         pMinCompareValue;   /*  The default value to be used for comparisons. */
  PVOID                         pMaxCompareValue;   /*  The default value to be used for comparisons. */
  PSZ                           pszEditControlClass; /*  Window class to be used to edit the compare value. */
  PFNCOMPARE                    pfnSort;            /*  Sort function for this field. */
  PSZ                           pNewComp;           /*  Pointer to an array of strings containing a description
} CLASSFIELDINFO;

typedef CLASSFIELDINFO *PCLASSFIELDINFO;
```

The CLASSFIELDINFO data structure is similar to the FIELDINFO data structure of the container control, with the following differences:

- The *offStruct* field from the FIELDINFO data structure is changed to *ulReserved*.
- Compare, Sort, and Ownerdraw fields (starting at *pfnOwnerDraw* and continuing through the end of the structure) have been added so that object Details View fields can be customized and objects can be found or included by any of their displayable object Details View fields.
- The *offFieldData* and *ulLenFieldData* fields have been added.

 As an example of the use of these two fields, if an application has the following details data:

  ```
  typedef struct _SAMPLE {
     CDATE  cdate;
     CTIME  ctime;
     PSZ    psz;
  }  SAMPLE;
  ```

then

```
classfieldinfo[0].offFieldData   = FIELDOFFSET(SAMPLE,cdate);
classfieldinfo[0].ulLenFieldData = FIELDOFFSET(SAMPLE,ctime) -
                                   FIELDOFFSET(SAMPLE,cdate);

classfieldinfo[1].offFieldData   = FIELDOFFSET(SAMPLE,ctime);
classfieldinfo[1].ulLenFieldData = FIELDOFFSET(SAMPLE,psz) -
                                   FIELDOFFSET(SAMPLE,ctime);

classfieldinfo[2].offFieldData   = FIELDOFFSET(SAMPLE,psz);
classfieldinfo[2].ulLenFieldData = sizeof(SAMPLE) -
                                   FIELDOFFSET(SAMPLE,psz);

/* FIELDOFFSET is a macro that calculates the byte offset of a field *
 * within a structure.                                               */
```

Note: It is essential that the fields be linked in the order shown above-cdate must be followed by ctime, and finally psz. The *pfnCompare* and *pfnSort* fields generally point to the same function, but they can point to different functions if you want them to behave differently.

CLASSFIELDINFO Field - cb

cb (ULONG)
Size of the CLASSFIELDINFO structure.

CLASSFIELDINFO Field - flData

flData (ULONG)
Attributes of the field's data.

Possible values are shown in the following list:

- Specify one of the following for each column to choose the type of data that is displayed in each column:

 CFA_BITMAPORICON
 The column contains bit-map or icon data.

 CFA_DATE
 The data in the column is displayed in date format. National Language Support (NLS) is enabled for date format. Use the data structure described in CDATE

 CFA_STRING
 Character or text data is displayed in this column.

 CFA_TIME
 The data in the column is displayed in time format. National Language Support (NLS) is enabled for time format. Use the data structure described in CTIME.

 CFA_ULONG
 Unsigned number data is displayed in this column. National Language Support (NLS) is enabled for number format.

- Specify any or all of the following column attributes:

 CFA_FIREADONLY
 Prevents text in a FIELDINFO data structure (text in a column) from being edited directly. This attribute applies only to columns for which the CFA_STRING attribute has been specified.

 CFA_HORZSEPARATOR
 A horizontal separator is provided beneath column headings.

 CFA_INVISIBLE

Invisible container column. The default is visible.

CFA_OWNER
Ownerdraw is enabled for this container column.

CFA_SEPARATOR
A vertical separator is drawn after this column.

- Specify one of the following for each column to vertically position data in that column:

 CFA_BOTTOM
 Bottom-justifies field data.

 CFA_TOP
 Top-justifies field data.

 CFA_VCENTER
 Vertically centers field data. This is the default.

- Specify one of the following for each column to horizontally position data in that column. These attributes can be combined with the attributes used for vertical positioning of column data by using an OR operator (|).

 CFA_CENTER
 Horizontally centers field data.

 CFA_LEFT
 Left-justifies field data. This is the default.

 CFA_RIGHT
 Right-justifies field data.

- Specify the following if new comparison descriptions are used:

 CFA_NEWCOMP
 Tells the system to use strings specified in *pNewComp* .

- Specify the following if a pointer to an object needs to be passed as the first parameter of a user-defined comparison function:

 CFA_OBJECT
 Tells the system that the applications wants to use its own comparison function in which the first parameter is a pointer to an object. For example:

  ```
  LONG MyComp(WPObject *obj, PSZ str2)
  ```

 This flag must be ORed (|) together with one of the type flags (CFA_BITMAPORICON, CFA_DATE, CFA_STRING, CFA_TIME, or CFA_ULONG) to tell the object the type that is being compared.

CLASSFIELDINFO Field - flTitle

flTitle (ULONG)
Attributes of the field's title.

Possible values are described in the following list:

- Specify the following if icon or bit-map data is to be displayed in the column heading:

 CFA_BITMAPORICON
 The column heading contains icon or bit-map data. If CFA_BITMAPORICON is not specified, any data that is assigned to a column heading is assumed to be character or text data.

- Specify the following to prevent direct editing of a column heading:

 CFA_FITITLEREADONLY
 Prevents a column heading from being edited directly.

- Specify one of the following for each column heading to vertically position data in that column heading:

 CFA_TOP

Top-justifies column headings.

CFA_BOTTOM
Bottom-justifies column headings.

CFA_VCENTER
Vertically centers column headings. This is the default.

- Specify one of the following for each column heading to horizontally position data in that column heading. These attributes can be combined with the attributes used for vertical positioning of column heading data by using an OR operator (|).

 CFA_CENTER
 Horizontally centers column headings.

 CFA_LEFT
 Left-justifies column headings. This is the default.

 CFA_RIGHT
 Right-justifies column headings.

CLASSFIELDINFO Field - pTitleData

pTitleData (PVOID)
Title data.

Default is string. If the *flTitle* field is set to the CFA_BITMAPORICON attribute, this must be an icon or bit-map handle.

CLASSFIELDINFO Field - ulReserved

ulReserved (ULONG)
Reserved.

CLASSFIELDINFO Field - pUserData

pUserData (PVOID)
Pointer to user data.

CLASSFIELDINFO Field - pNextFieldInfo

pNextFieldInfo (struct _CLASSFIELDINFO *)
Pointer to the next linked CLASSFIELDINFO structure.

CLASSFIELDINFO Field - cxWidth

cxWidth (ULONG)
Width of the field in pels.

CLASSFIELDINFO Field - offFieldData

offFieldData (ULONG)
Offset from beginning of this class's data for this field.

The first field is zero.

CLASSFIELDINFO Field - ulLenFieldData

ulLenFieldData (ULONG)
Width of data in bytes.

Four bytes for a pointer.

CLASSFIELDINFO Field - pfnOwnerDraw

pfnOwnerDraw (PFNOWNDRW)
Ownerdraw procedure for Details View column.

You would use this if you wanted to draw the field in the Details View of your object. An example of this might be if you wanted to draw a graphic image in the field.

This field can be NULL except if ownerdraw is being used (that is, if the object is choosing to draw the field itself).

CLASSFIELDINFO Field - flCompare

flCompare (ULONG)
Flags.

Possible values are:

COMPARE_SUPPORTED
Your field can be compared.
SORTBY_SUPPORTED
Your field can be sorted on.

CLASSFIELDINFO Field - pfnCompare

pfnCompare (PFNCOMPARE)
Pointer to a comparison function.

A comparison function is used when the object is doing its own Include Page filtering (in the Settings notebook).

For example, if a string details field exists, the field could be set to pfnCompare = StringCompare, where StringCompare is defined as:

```
LONG StringCompare(PSZ str1, PZS str2)
```

This field can be NULL unless ownerdraw is used.

CLASSFIELDINFO Field - DefaultComparison

DefaultComparison (ULONG)
Default comparison operator in the Include Page Criteria dialog box (in the Settings notebook).

For example, CMP_LESS_OR_EQUAL or CMP_NOT_EQUAL.

CLASSFIELDINFO Field - ulLenCompareValue

ulLenCompareValue (ULONG)
Maximum length of the compare data.

Can be NULL unless ownerdraw is used.

CLASSFIELDINFO Field - pDefCompareValue

pDefCompareValue (PVOID)
The default value to be used for comparisons.

Can be NULL unless ownerdraw is used.

CLASSFIELDINFO Field - pMinCompareValue

pMinCompareValue (PVOID)
The default value to be used for comparisons.

Can be NULL unless ownerdraw is used.

CLASSFIELDINFO Field - pMaxCompareValue

pMaxCompareValue (PVOID)
The default value to be used for comparisons.

Can be NULL unless ownerdraw is used.

CLASSFIELDINFO Field - pszEditControlClass

pszEditControlClass (PSZ)
Window class to be used to edit the compare value.

Can be NULL unless ownerdraw is used.

CLASSFIELDINFO Field - pfnSort

pfnSort (PFNCOMPARE)
Sort function for this field.

This function is used when the object is doing its own folder sorting.

Can be NULL unless ownerdraw is used.

CLASSFIELDINFO Field - pNewComp

pNewComp (PSZ)
Pointer to an array of strings containing a description of a comparison type.

The last element of the string must be NULL.

If new comparison types are added, code must be provided for the comparison function.

CNRINFO

Structure that contains information about the container.

```
typedef struct _CNRINFO {
  ULONG          cb;                 /*  Structure size. */
  PVOID          pSortRecord;        /*  Pointer to the comparison function for sorting container records, or N
  PFIELDINFO     pFieldInfoLast;     /*  Pointer to the last column in the left window of the split details vie
  PFIELDINFO     pFieldInfoObject;   /*  Pointer to a column that represents an object in the details view. */
  PSZ            pszCnrTitle;        /*  Title text, or NULL. */
  ULONG          flWindowAttr;       /*  Window attributes. */
  POINTL         ptlOrigin;          /*  Workspace origin. */
  ULONG          cDelta;             /*  Threshold. */
  ULONG          cRecords;           /*  Number of records. */
  SIZEL          slBitmapOrIcon;     /*  Icon/bit-map size. */
  SIZEL          slTreeBitmapOrIcon; /*  Icon/bit-map size. */
```

```
  HBITMAP        hbmExpanded;         /*  Bit-map handle. */
  HBITMAP        hbmCollapsed;        /*  Bit-map handle. */
  HPOINTER       hptrExpanded;        /*  Icon handle. */
  HPOINTER       hptrCollapsed;       /*  Icon handle. */
  LONG           cyLineSpacing;       /*  Vertical space. */
  LONG           cxTreeIndent;        /*  Horizontal space. */
  LONG           cxTreeLine;          /*  Line width. */
  ULONG          cFields;             /*  Number of columns. */
  LONG           xVertSplitbar;       /*  Split bar position. */
} CNRINFO;

typedef CNRINFO *PCNRINFO;
```

CNRINFO Field - cb

cb (ULONG)
Structure size.

The size (in bytes) of the CNRINFO data structure.

CNRINFO Field - pSortRecord

pSortRecord (PVOID)
Pointer to the comparison function for sorting container records, or NULL.

If NULL, which is the default condition, no sorting is performed. Sorting occurs only during record insertion and when changing the value of this field. The third parameter of the comparison function, *pStorage* , must be NULL. See "CM_SORTRECORD" in the *Presentation Manager Programming Reference* for a further description of the comparison function.

CNRINFO Field - pFieldInfoLast

pFieldInfoLast (PFIELDINFO)
Pointer to the last column in the left window of the split details view, or NULL.

The default is NULL, which causes all columns to be positioned in the left window.

CNRINFO Field - pFieldInfoObject

pFieldInfoObject (PFIELDINFO)
Pointer to a column that represents an object in the details view.

The data for this FIELDINFO structure must contain icons or bit maps. In-use emphasis is applied to this column of icons or bit maps only. The default is the leftmost column in the unsplit details view or the leftmost column in the left window of the split details view.

CNRINFO Field - pszCnrTitle

pszCnrTitle (PSZ)
Title text, or NULL.

Text for the container title. The default is NULL.

CNRINFO Field - flWindowAttr

flWindowAttr (ULONG)
Window attributes.

Consists of container window attributes.

- Specify one of the following container views, which determine the presentation format of items in a container:

 CV_ICON
 In the icon view, the container items are represented as icon/text or bit-map/text pairs, with text beneath the icons or bit maps. This is the default view. This view can be combined with the CV_MINI style bit by using an OR operator (|).

 CV_NAME
 In the name view, the container items are represented as icon/text or bit-map/text pairs, with text to the right of the icons or bit maps. This view can be combined with the CV_MINI and CV_FLOW style bits by using OR operators (|).

 CV_TEXT
 In the text view, the container items are displayed as a list of text strings. This view can be combined with the CV_FLOW style bit by using an OR operator (|).

 CV_TREE
 In the tree view, the container items are represented in a hierarchical manner. The tree view has three forms, which are defined in the following list. If you specify CV_TREE by itself, the tree icon view is used.

 - Tree icon view

 The tree icon view is specified by using a logical OR operator to combine the tree view with the icon view (CV_TREE | CV_ICON). Container items in this view are represented as icon/text pairs or bit-map/text pairs, with text to the right of the icons or bit maps. Also, a collapsed or expanded icon or bit map is displayed to the left of parent items. If this icon or bit map is a *collapsed* icon or bit map, selecting it will cause the parent item to be expanded so that its child items are displayed below it. If this icon or bit map is an *expanded* icon or bit map, selecting it will cause the parent's child items to be removed from the display. The default collapsed and expanded bit maps provided by the container use a plus sign (+) and a minus sign (-), respectively, to indicate that items can be added to or subtracted from the display.

 - Tree name view

 The tree name view is specified by using a logical OR operator to combine the tree view with the name view (CV_TREE | CV_NAME). Container items in this view are displayed as either icon/text pairs or bit-map/text pairs, with text to the right of the icons or bit maps. However, the indicator that represents whether an item can be collapsed or expanded, such as a plus or minus sign, is included in the icon or bit map that represents that item, not in a separate icon or bit map as in the tree icon and tree text views. The container control does not provide default collapsed and expanded bit maps for the tree name view.

 - Tree text view

 The tree text view is specified by using a logical OR operator to combine the tree view with the text view (CV_TREE | CV_TEXT). Container items in this view are displayed as a list of text strings. As in the tree icon view, a collapsed or expanded icon or bit map is displayed to the left of parent items.

 CV_DETAIL

In the details view, the container items are presented in columns. Each column can contain icons or bit maps, text, numbers, dates, or times.

- Specify one or both of the following view styles by using an OR operator (|) to combine them with the specified view. These view styles are optional.

 CV_MINI
 Produces a mini-icon whose size is based on the Presentation Manager (PM) SV_CYMENU system value to produce a device-dependent mini-icon.

 The CV_MINI view style bit is ignored when:

 - The text view (CV_TEXT), tree view (CV_TREE), or details view (CV_DETAIL) are displayed
 - The CCS_MINIRECORDCORE style bit is specified.

 If this style bit is not specified and the icon view (CV_ICON) or name view (CV_NAME) is used, the default, regular-sized icon is used. The size of regular-sized icons is based on the value in the *slBitmapOrIcon* field of the CNRINFO data structure. If this field is equal to 0, the PM SV_CXICON and SV_CYICON system values for width and height, respectively, are used. Icon sizes are consistent with PM-defined icon sizes for all devices.

 CV_FLOW
 Dynamically arranges container items in columns in the name and text views. These are called flowed name and flowed text views. If this style bit is set for the name view (CV_NAME) or text view (CV_TEXT), the container items are placed in a single column until the bottom of the client area is reached. The next container item is placed in the adjacent column to the right of the filled column. This process is repeated until all of the container items are positioned in the container. The width of each column is determined by the longest text string in that column. The size of the window determines the depth of the client area.

 If this style bit is not specified, the default condition for the name and text views is to vertically fill the container in a single column without flowing the container items. If this style bit is set for the icon view (CV_ICON) or details view (CV_DETAIL), it is ignored.

- Specify either of the following to indicate whether the container will display icons or bit maps:

 CA_DRAWICON
 Icons are used for the icon, name, tree, or details views. This is the default. This container attribute should be used with the *hptrIcon* and *hptrMiniIcon* fields of the RECORDCORE data structure.

 CA_DRAWBITMAP
 Bit maps are used for the icon, name, tree, or details views. This container attribute can be used with the *hbmBitmap* and *hbmMiniBitmap* fields of the RECORDCORE data structure.

 Notes

 1. If both the CA_DRAWICON and CA_DRAWBITMAP attributes are specified, the CA_DRAWICON attribute is used.
 2. If the CCS_MINIRECORDCORE style bit is specified when a container is created, the *hptrIcon* field of the MINIRECORDCORE data structure is used.

- Specify one of the following attributes to provide target emphasis for the name, text, and details views. If neither ordered nor mixed target emphasis is specified, the emphasis is drawn around the record.

 CA_ORDEREDTARGETEMPH
 Shows where a container record can be dropped during direct manipulation by drawing a line beneath the record. Ordered target emphasis does not apply to the icon and tree views.

 CA_MIXEDTARGETEMPH
 Shows where a container record can be dropped during direct manipulation either by drawing a line between two items or by drawing lines around the container record. Mixed target emphasis does not apply to the icon and tree views.

- Specify the following attribute to draw lines that show the relationship between items in the tree view.

 CA_TREELINE
 Shows the relationship between all items in the tree view.

- Specify the following to draw container records, paint the background of the container, or both:

 CA_OWNERDRAW
 Ownerdraw for the container, which allows the application to draw container records.

 CA_OWNERPAINTBACKGROUND
 Allows the application to subclass the container and paint the background. If specified, and the container

is subclassed, the application receives the CM_PAINTBACKGROUND message in the subclass procedure. Otherwise, the container paints the background using the color specified by SYSCLR_WINDOW, which can be changed by using the PP_BACKGROUNDCOLOR or PP_BACKGROUNDCOLORINDEX presentation parameter in the WM_PRESPARAMCHANGED (in Container Controls)

- Specify the following if the container is to have a title:

 CA_CONTAINERTITLE
 Allows you to include a container title. The default is no container title.

- Specify one or both of the following container title attributes. These are valid only if the CA_CONTAINERTITLE attribute is specified.

 CA_TITLEREADONLY
 Prevents the container title from being edited directly. The default is to allow the container title to be edited.

 CA_TITLESEPARATOR
 Puts a separator line between the container title and the records beneath it. The default is no separator line.

- Specify one of the following to position the container title. These are valid only if the CA_CONTAINERTITLE attribute is specified.

 CA_TITLECENTER
 Centers the container title. This is the default.

 CA_TITLELEFT
 Left-justifies the container title.

 CA_TITLERIGHT
 Right-justifies the container title.

- Specify the following to display column headings in the details view:

 CA_DETAILSVIEWTITLES
 Allows you to include column headings in the details view. The default is no column headings.

CNRINFO Field - ptlOrigin

ptlOrigin (POINTL)
Workspace origin.

Lower-left origin of the workspace in virtual coordinates, used in the icon view. The default origin is (0,0).

CNRINFO Field - cDelta

cDelta (ULONG)
Threshold.

An application-defined threshold, or number of records, from either end of the list of available records. Used when a container needs to handle large amounts of data. The default is 0. See the *OS/2 Programming Guide* for more information about specifying deltas.

CNRINFO Field - cRecords

cRecords (ULONG)
Number of records.

The number of records in the container. Initially, this field is 0.

CNRINFO Field - slBitmapOrIcon

slBitmapOrIcon (SIZEL)
Icon/bit-map size.

The size (in pels) of icons or bit maps. The default is the system size.

CNRINFO Field - slTreeBitmapOrIcon

slTreeBitmapOrIcon (SIZEL)
Icon/bit-map size.

The size (in pels) of the expanded and collapsed icons or bit maps used in the tree icon and tree text views.

CNRINFO Field - hbmExpanded

hbmExpanded (HBITMAP)
Bit-map handle.

The handle of the bit map to be used to represent an expanded parent item in the tree icon and tree text views. If neither an icon handle (see *hptrExpanded*) nor a bit-map handle is specified, a default bit map with a minus sign (-) is provided.

CNRINFO Field - hbmCollapsed

hbmCollapsed (HBITMAP)
Bit-map handle.

The handle of the bit map to be used to represent a collapsed parent item in the tree icon and tree text views. If neither an icon handle (see *hptrCollapsed*) nor a bit-map handle is specified, a default bit map with a plus sign (+) is provided.

CNRINFO Field - hptrExpanded

hptrExpanded (HPOINTER)
Icon handle.

The handle of the icon to be used to represent an expanded parent item in the tree icon and tree text views. If neither an icon handle nor a bit-map handle (see *hbmExpanded*) is specified, a default bit map with a minus sign (-) is provided.

CNRINFO Field - hptrCollapsed

hptrCollapsed (HPOINTER)
Icon handle.

The handle of the icon to be used to represent a collapsed parent item in the tree icon and tree text views. If neither an icon handle nor a bit-map handle (see *hbmCollapsed*) is specified, a default bit map with a plus sign (+) is provided.

CNRINFO Field - cyLineSpacing

cyLineSpacing (LONG)
Vertical space.

The amount of vertical space (in pels) between the records. If you specify a value that is less than 0, a default value is used.

CNRINFO Field - cxTreeIndent

cxTreeIndent (LONG)
Horizontal space.

The amount of horizontal space (in pels) between levels in the tree view. If you specify a value that is less than 0, a default value is used.

CNRINFO Field - cxTreeLine

cxTreeLine (LONG)
Line width.

The width of the lines (in pels) that show the relationship between tree items. If you specify a value that is less than 0, a default value is used. Also, if the CA_TREELINE container attribute of the *flWindowAttr* field is not specified, these lines are not drawn.

CNRINFO Field - cFields

cFields (ULONG)
Number of columns.

The number of FIELDINFO structures in the container. Initially, this field is 0.

CNRINFO Field - xVertSplitbar

xVertSplitbar (LONG)
Split bar position.

The initial position of the split bar relative to the container used in the details view. If this value is less than 0, the split bar is not used. The default value is negative one (-1).

CRITERIA

Details comparison criteria structure.

```
typedef struct _CRITERIA {
  PVOID                 pReserved;            /*  Reserved.  Set to NULL. */
  struct _CRITERIA      *pNext;               /*  Pointer to the next criteria record. */
  struct _CRITERIA      *pPrev;               /*  Pointer to the previous criteria record. */
  ULONG                 ClassFieldInfoIndex;  /*  Index of the detaild field. */
  ULONG                 LogicalOperator;      /*  CRITERIA_AND or CRITERIA_OR */
  ULONG                 ComparisonType;       /*  The comparison type. */
  ULONG                 ulLenCompareValue;    /*  Sizeof(ComparisonData). */
  CHAR                  CompareValue[1];      /*  Value to be compared with. */
} CRITERIA;

typedef CRITERIA *PCRITERIA;
```

CRITERIA Field - pReserved

pReserved (PVOID)
Reserved. Set to NULL.

CRITERIA Field - pNext

pNext (struct _CRITERIA *)
Pointer to the next criteria record.

CRITERIA Field - pPrev

pPrev (struct _CRITERIA *)
Pointer to the previous criteria record.

CRITERIA Field - ClassFieldInfoIndex

ClassFieldInfoIndex (ULONG)
Index of the detaild field.

CRITERIA Field - LogicalOperator

LogicalOperator (ULONG)
CRITERIA_AND or CRITERIA_OR

CRITERIA Field - ComparisonType

ComparisonType (ULONG)
The comparison type.

May be one of the following values:

- CMP_EQUAL
- CMP_GREATER
- CMP_LESS
- CMP_GREATER_OR_EQUAL
- CMP_LESS_OR_EQUAL
- CMP_NO_EQUAL

CRITERIA Field - ulLenCompareValue

ulLenCompareValue (ULONG)
Sizeof(ComparisonData).

CRITERIA Field - CompareValue[1]

CompareValue[1] (CHAR)
Value to be compared with.

CTIME

Structure that contains time information for a data element in the details view of a container control.

```
typedef struct _CTIME {
  UCHAR     hours;       /*  Current hour. */
  UCHAR     minutes;     /*  Current minute. */
  UCHAR     seconds;     /*  Current second. */
  UCHAR     ucReserved;  /*  Reserved. */
} CTIME;

typedef CTIME *PCTIME;
```

CTIME Field - hours

hours (UCHAR)
Current hour.

CTIME Field - minutes

minutes (UCHAR)
Current minute.

CTIME Field - seconds

seconds (UCHAR)
Current second.

CTIME Field - ucReserved

ucReserved (UCHAR)
Reserved.

DEVOPENSTRUC

Open-device data structure.

```
typedef struct _DEVOPENSTRUC {
  PSZ           pszLogAddress;       /*  Logical address. */
  PSZ           pszDriverName;       /*  Driver name. */
  PDRIVDATA     pdriv;               /*  Driver data. */
  PSZ           pszDataType;         /*  Data type. */
```

```
  PSZ           pszComment;          /*  Comment. */
  PSZ           pszQueueProcName;    /*  Queue-processor name. */
  PSZ           pszQueueProcParams;  /*  Queue-processor parameters. */
  PSZ           pszSpoolerParams;    /*  Spooler parameters. */
  PSZ           pszNetworkParams;    /*  Network parameters. */
} DEVOPENSTRUC;

typedef DEVOPENSTRUC *PDEVOPENSTRUC;
```

DEVOPENSTRUC Field - pszLogAddress

pszLogAddress (PSZ)
Logical address.

This is required for an OD_DIRECT device being opened with DevOpenDC; it is the logical device address, such as "LPT1" on OS/2. Some drivers may accept a file name for this parameter or even a named pipe.

Where output is to be queued (for an OD_QUEUED device), this is the name of the queue for the output device. The queue name can be a UNC name.

Note: This parameter can be a port name for a printer device context.

DEVOPENSTRUC Field - pszDriverName

pszDriverName (PSZ)
Driver name.

Character string identifying the printer driver, for example, LASERJET. The Default Device Driver field of the PRQINFO3 structure, associated with the required print queue, gives the driver and device name, separated by a period, for example LASERJET.HP LaserJet IIID. It can contain only the name up to the period, for example LASERJET.

DEVOPENSTRUC Field - pdriv

pdriv (PDRIVDATA)
Driver data.

Data that is to be passed directly to the PM device driver. Whether any of this is required depends upon the device driver.

For printer device context, this is a pointer to the job properties data.

DEVOPENSTRUC Field - pszDataType

pszDataType (PSZ)
Data type.

For an OD_QUEUED or OD_DIRECT device, this parameter defines the type of data that is to be queued as follows:

PM_Q_STD	Standard format
PM_Q_RAW	Raw format

Note that a device driver can define other data types.

For OD_QUEUED or OD_DIRECT defice types, the default is supplied by the device driver if *pszDataType* is not specified. For any other device type, *pszDataType* is ignored.

DEVOPENSTRUC Field - pszComment

pszComment (PSZ)
Comment.

Optional character string that the printer object displays to the user in a job settings notebook. It is recommended that the application include its own name in this comment string.

Note: The job title text is derived from the document name passed to DevEscape (DEVESC_STARTDOC).

DEVOPENSTRUC Field - pszQueueProcName

pszQueueProcName (PSZ)
Queue-processor name.

This is the name of the queue processor, for queued output, and is usually the default.

DEVOPENSTRUC Field - pszQueueProcParams

pszQueueProcParams (PSZ)
Queue-processor parameters.

Queue processor parameters (optional). They can include information such as the number of copies you want to print and the size of the output area on the printed page.

The first parameter (*COP*) is used for all spool-file formats. The remaining parameters are valid for PM_Q_STD spool files only. Because PM_Q_STD data are used mainly for *graphic* data, these parameters are described in relation to the printing of picture files.

The PMPRINT/PMPLOT queue-processor parameters are separated by spaces and are:

COP=**n**
The *COP* parameter specifies the number of copies of the spool file that you want printed. The value of *n* must be an integer in the range of 1 through 999.

The default is *COP=1* .

ARE=C | **w,h,l,t**
The *ARE* parameter determines the size and position of the output area. This is the area of the physical page to which printing is restricted.

The default value of *ARE=C* means that the output area is the whole page. Note, however, that the printer cannot print outside its own device clip limits.

To size and position the output area at a specific point on the page, use *ARE=* **w,h,l,t**, where:

w, h — are the width and height of the desired output area.

l, t — are the offsets of the upper-left corner of the output area from the left (l) and from the top (t) of the maximum output area.

These four values must be given as percentages of the maximum output dimensions. The maximum output area is the area within the device clip limits.

FIT=S | **l,t**

The *FIT* parameter determines which part of the picture is to be printed. You can request the whole of the picture, scaled to fit the output area; or you can position the picture (actual size) anywhere within the output area. This could mean that the picture is clipped at the boundaries of the output area.

The default value of *FIT=S* causes the output to be scaled until the larger of the height or width just fits within the defined output area. The aspect ratio of the picture is maintained.

To print the picture in actual size, use *FIT=***l,t**, where *l,t* are the coordinates of the point in the picture that you want positioned at the center of the output area: *l* is measured from the left edge of the picture; and *t* is measured from the top edge. The coordinates must be given as percentages of the actual dimensions of the picture.

XFM=0 | 1

The *XFM* parameter enables you to override the picture-positioning and clipping instructions that are provided by the *ARE* and *FIT* parameters, including their defaults.

The default value of *XFM=1* allows the appearance of the output to be determined by the settings of the *ARE* and *FIT* parameters.

A value of *XFM=0* yields output as specified in the picture file. For example, applications that use many different forms can define different positions on each form for their output.

COL=M | C

The *COL* parameter enables you to specify color output if you have a color printer.

A value of *COL=M* creates monochrome output (black foreground with no background color). This is supported by all devices.

A value of *COL=C* creates color output. If you request color output on a monochrome device, the printer presentation driver tries to satisfy your request, which can cause problems because the only color available is black. For example, if the picture file specifies a red line on a blue background, both are drawn in black.

The default is *COL=M* when you are addressing a monochrome printer and *COL=C* when you are addressing a color printer.

MAP=N | A

The *MAP* parameter enables you to decide how the *neutral* colors (those that are not specified in the picture file) are printed.

The default value of *MAP=N* yields a *normal* representation of the screen picture on a printed page, which means that the page background is white and the foreground is black.

A value of *MAP=A* provides the reverse of the normal representation: the background is black and the foreground is white on the printed page.

CDP=codepage

The *CDP* parameter overrides the code page to being used for PM_Q_RAW print jobs. The print queue driver uses DEVESC_SETMODE to set the code page, but not all printer drivers support this device escape.

XLT=0 | 1

The *XLT* parameter can eliminate the translation component when printing a metafile if *XLT=1* .

When the resolution of the device is higher than that of the world coordinate space, a small translation of world coordinate point (0,0) occurs on the device to preserve the accuracy of the mapping from world to device coordinate units. For example, (0,0) becomes (1,1) if there are 3 pels to every world coordinate.

Normally, this is not noticeable, but it can be a problem with some devices. For example, in order to draw a complete row of 80 characters using a device font, a device may require the text to start at device coordinate position zero. Starting at a position other than zero may cause one or more characters at the end of the row to be clipped. In such cases, elimination of the translation is important and can be accomplished by specifying *XLT=1* .

The default is *XLT=0* .

DEVOPENSTRUC Field - pszSpoolerParams

pszSpoolerParams (PSZ)
Spooler parameters.

Spooler parameters (optional) are separated by spaces. They are used for scheduling print jobs and are as follows:

- The form names that identify the paper to be used, for example, FORM=A4,A5,ENV. The form names are optional; but if they are provided, the spooler is able to hold off printing the jobs until the required form is installed in the printer. If the form name is not provided, the spooler attempts to print the job. The printer driver recognizes that there is a forms problem and displays a FORMS MISMATCH message box.
- Priority of the print job, for example, *PRTY=60*. The priority is specified as an integer in the range 1 through 99; 99 is the highest. The default priority value is 50. The application can use the spooler priority parameter to prioritize its own jobs; however, it is not good practice for an application always to use priority 99 in an attempt to get its jobs printed first.

DEVOPENSTRUC Field - pszNetworkParams

pszNetworkParams (PSZ)
Network parameters.

Optional parameter that can be used to specify network options; for example, *USER=JOESMITH*.

DRAGIMAGE

Dragged-object-image structure which describes the images that are to be drawn under the direct-manipulation pointer for the duration of a drag operation.

```
typedef struct _DRAGIMAGE {
  USHORT      cb;           /*  Size, in bytes, of the DRAGIMAGE structure. */
  USHORT      cptl;         /*  The number of points in the point array if fl is specified as DRG_POLYGON. */
  LHANDLE     hImage;       /*  Handle representing the image to display. */
  SIZEL       sizlStretch;  /*  Dimensions for stretching when fl is specified as DRG_STRETCH. */
  ULONG       fl;           /*  Flags. */
  SHORT       cxOffset;     /*  X-offset from the pointer hot spot to the origin of the image. */
  SHORT       cyOffset;     /*  Y-offset from the pointer hot spot to the origin of the image. */
} DRAGIMAGE;

typedef DRAGIMAGE *PDRAGIMAGE;
```

DRAGIMAGE Field - cb

cb (USHORT)
Size, in bytes, of the DRAGIMAGE structure.

DRAGIMAGE Field - cptl

cptl (USHORT)

The number of points in the point array if *fl* is specified as DRG_POLYGON.

DRAGIMAGE Field - hImage

hImage (LHANDLE)
Handle representing the image to display.

The type is determined by *fl*.

DRAGIMAGE Field - sizlStretch

sizlStretch (SIZEL)
Dimensions for stretching when *fl* is specified as DRG_STRETCH.

DRAGIMAGE Field - fl

fl (ULONG)
Flags.

DRG_ICON
hImage is an HPOINTER.
DRG_BITMAP
hImage is an HBITMAP.
DRG_POLYGON
hImage is a pointer to an array of points that will be connected with GpiPolyLine to form a polygon. The first point of the array should be (0,0), and the other points should be placed relative to this position.
DRG_STRETCH
If DRG_ICON or DRG_BITMAP is specified, the image is expanded or compressed to the dimensions specified by *sizlStretch* .
DRG_TRANSPARENT
If DRG_ICON is specified, an outline of the icon is generated and displayed instead of the original icon.
DRG_CLOSED
If DRG_POLYGON is specified, a closed polygon is formed by moving the current position to the last point in the array before calling GpiPolyLine.

DRAGIMAGE Field - cxOffset

cxOffset (SHORT)
X-offset from the pointer hot spot to the origin of the image.

DRAGIMAGE Field - cyOffset

cyOffset (SHORT)
Y-offset from the pointer hot spot to the origin of the image.

DRAGINFO

Drag-information structure.

```
typedef struct _DRAGINFO {
  ULONG      cbDraginfo;   /*  Structure size, in bytes. */
  USHORT     cbDragitem;   /*  Size, in bytes, of each DRAGITEM structure. */
  USHORT     usOperation;  /*  Modified drag operations. */
  HWND       hwndSource;   /*  Window handle of the source of the drag operation. */
  SHORT      xDrop;        /*  X-coordinate of drop point expressed in desktop coordinates. */
  SHORT      yDrop;        /*  Y-coordinate of drop point expressed in desktop coordinates. */
  USHORT     cditem;       /*  Count of DRAGITEM structures. */
  USHORT     usReserved;   /*  Reserved. */
} DRAGINFO;

typedef DRAGINFO *PDRAGINFO;
```

DRAGINFO Field - cbDraginfo

cbDraginfo (ULONG)
Structure size, in bytes.

The size includes the array of DRAGITEM structures.

DRAGINFO Field - cbDragitem

cbDragitem (USHORT)
Size, in bytes, of each DRAGITEM structure.

DRAGINFO Field - usOperation

usOperation (USHORT)
Modified drag operations.

An application can define its own modified drag operations for use when simulating a drop. These operations must have a value greater than DO_UNKNOWN. Possible values are described in the following list:

DO_DEFAULT
: Execute the default drag operation. No modifier keys are pressed.

DO_COPY
: Execute a copy operation. The **Ctrl** key is pressed.

DO_LINK
: Execute a link operation. The **Ctrl+Shift** keys are pressed.

DO_MOVE	Execute a move operation. The **Shift** key is pressed.
DO_CREATE	Execute a create operation. (A template is being dropped.)
DO_NEW	Execute a create another operation. This value should be defined as DO_UNKNOWN+3 if it is not recognized in the current level of the toolkit.
DO_UNKNOWN	An undefined combination of modifier keys is pressed.

DRAGINFO Field - hwndSource

hwndSource (HWND)
Window handle of the source of the drag operation.

DRAGINFO Field - xDrop

xDrop (SHORT)
X-coordinate of drop point expressed in desktop coordinates.

DRAGINFO Field - yDrop

yDrop (SHORT)
Y-coordinate of drop point expressed in desktop coordinates.

DRAGINFO Field - cditem

cditem (USHORT)
Count of DRAGITEM structures.

DRAGINFO Field - usReserved

usReserved (USHORT)
Reserved.

DRAGITEM

Drag-object structure.

```
typedef struct _DRAGITEM {
  HWND        hwndItem;           /*  Window handle of the source of the drag operation. */
  ULONG       ulItemID;           /*  Information used by the source to identify the object being dragged. */
  HSTR        hstrType;           /*  String handle of the object type. */
  HSTR        hstrRMF;            /*  String handle of the rendering mechanism and format. */
  HSTR        hstrContainerName;  /*  String handle of the name of the container holding the source object. */
  HSTR        hstrSourceName;     /*  String handle of the name of the source object. */
  HSTR        hstrTargetName;     /*  String handle of the suggested name of the object at the target. */
  SHORT       cxOffset;           /*  X-offset from the pointer hot spot to the origin of the image that represen
  SHORT       cyOffset;           /*  Y-offset from the pointer hot spot to the origin of the image that represen
  USHORT      fsControl;          /*  Source-object control flags. */
  USHORT      fsSupportedOps;     /*  Direct manipulation operations supported by the source object. */
} DRAGITEM;

typedef DRAGITEM *PDRAGITEM;
```

DRAGITEM Field - hwndItem

hwndItem (HWND)
Window handle of the source of the drag operation.

DRAGITEM Field - ulItemID

ulItemID (ULONG)
Information used by the source to identify the object being dragged.

DRAGITEM Field - hstrType

hstrType (HSTR)
String handle of the object type.

The string handle must be created using the DrgAddStrHandle function. The string is of the form:

```
type[,type...]
```

The first type in the list must be the true type of the object. The following types are used by the OS/2* shell:

DRT_ASM
: Assembler code

DRT_BASIC
: BASIC code

DRT_BINDATA
: Binary data

DRT_BITMAP
: Bit map

DRT_C
: C code

DRT_COBOL
: COBOL code

DRT_DLL
: Dynamic link library

DRT_DOSCMD
: DOS command file

DRT_EXE
: Executable file

DRT_FONT
: Font

DRT_FORTRAN
: FORTRAN code

DRT_ICON
: Icon

DRT_LIB
: Library

DRT_METAFILE
: Metafile

DRT_OS2CMD
: OS/2 command file

DRT_PASCAL
: Pascal code

DRT_RESOURCE
: Resource file

DRT_TEXT
: Text

DRT_UNKNOWN
: Unknown type.

DRAGITEM Field - hstrRMF

hstrRMF (HSTR)
: String handle of the rendering mechanism and format.

The string handle must be created using the DrgAddStrHandle function. The string is of the form:

```
mechfmt[,mechfmt...]
```

where mechfmt can be in either of the following formats:

- <mechanism(1),format(1)>
- (mechanism(1)[, mechanism(n)...]), (format(1)[,format(n)...])

The first mechanism/format pair must be the native rendering mechanism and format of the object.

Valid mechanisms are:

"DRM_DDE"	Dynamic data exchange
"DRM_OBJECT"	Item being dragged is a Workplace Shell object
"DRM_OS2FILE"	OS/2 file
"DRM_PRINT"	Object can be printed using direct manipulation.

Valid formats are:

"DRF_BITMAP"	OS/2 bit map
"DRF_DIB"	DIB
"DRF_DIF"	DIF
"DRF_DSPBITMAP"	Stream of bit-map bits
"DRF_METAFILE"	Metafile
"DRF_OEMTEXT"	OEM text

"DRF_OWNERDISPLAY"	Bit stream
"DRF_PTRPICT"	Printer picture
"DRF_RTF"	Rich text
"DRF_SYLK"	SYLK
"DRF_TEXT"	Null-terminated string
"DRF_TIFF"	TIFF
"DRF_UNKNOWN"	Unknown format.

DRAGITEM Field - hstrContainerName

hstrContainerName (HSTR)
String handle of the name of the container holding the source object.

The string handle must be created using the DrgAddStrHandle function.

DRAGITEM Field - hstrSourceName

hstrSourceName (HSTR)
String handle of the name of the source object.

The string handle must be created using the DrgAddStrHandle function.

DRAGITEM Field - hstrTargetName

hstrTargetName (HSTR)
String handle of the suggested name of the object at the target.

It is the responsibility of the source of the drag operation to create this string handle before calling DrgDrag.

DRAGITEM Field - cxOffset

cxOffset (SHORT)
X-offset from the pointer hot spot to the origin of the image that represents this object.

This value is copied from *cxOffset* in the DRAGIMAGE structure by DrgDrag.

DRAGITEM Field - cyOffset

cyOffset (SHORT)
Y-offset from the pointer hot spot to the origin of the image that represents this object.

This value is copied from *cyOffset* in the DRAGIMAGE structure by DrgDrag.

DRAGITEM Field - fsControl

fsControl (USHORT)
Source-object control flags.

DC_OPEN
Object is open
DC_REF
Reference to another object
DC_GROUP
Group of objects
DC_CONTAINER
Container of other objects
DC_PREPARE
Source requires a DM_RENDERPREPARE message before it establishes a data transfer conversation
DC_REMOVEABLEMEDIA
Object is on removable media, or object cannot be recovered after a move operation.

DRAGITEM Field - fsSupportedOps

fsSupportedOps (USHORT)
Direct manipulation operations supported by the source object.

DO_COPYABLE
Source supports DO_COPY
DO_LINKABLE
Source supports DO_LINK
DO_MOVEABLE
Source supports DO_MOVE.

DRAGTRANSFER

Drag-conversation structure.

```
typedef struct _DRAGTRANSFER {
  ULONG          cb;                /*  Size, in bytes, of the structure. */
  HWND           hwndClient;        /*  Handle of the client window. */
  PDRAGITEM      pditem;            /*  Pointer to the DRAGITEM structure that is to be rendered. */
  HSTR           hstrSelectedRMF;   /*  String handle for the selected rendering mechanism and format for the tra
  HSTR           hstrRenderToName;  /*  String handle representing the name where the source places, and the targ
  ULONG          ulTargetInfo;      /*  Reserved. */
  USHORT         usOperation;       /*  The operation. */
  USHORT         fsReply;           /*  Reply flags. */
} DRAGTRANSFER;

typedef DRAGTRANSFER *PDRAGTRANSFER;
```

DRAGTRANSFER Field - cb

cb (ULONG)
Size, in bytes, of the structure.

DRAGTRANSFER Field - hwndClient

hwndClient (HWND)
Handle of the client window.

This can be the target window or a window that represents an object in a container that was dropped on.

DRAGTRANSFER Field - pditem

pditem (PDRAGITEM)
Pointer to the DRAGITEM structure that is to be rendered.

This structure must exist within the DRAGINFO structure that was passed in the DM_DROP message.

DRAGTRANSFER Field - hstrSelectedRMF

hstrSelectedRMF (HSTR)
String handle for the selected rendering mechanism and format for the transfer operation.

This handle must be created using DrgAddStrHandle. The target is responsible for deleting this handle when the conversation is complete. The string is in the format: <mechanism,format>.

DRAGTRANSFER Field - hstrRenderToName

hstrRenderToName (HSTR)
String handle representing the name where the source places, and the target finds, the data that is rendered.

The target is responsible for deleting this string handle when the conversation terminates. The contents of this field vary according to the rendering mechanism. See the *hstrRMF* field in DRAGITEM.

OS/2 File	The string handle represents the fully qualified name of the file where the rendering will be placed.
DDE	This field is not used.
Print	This field is not used.

DRAGTRANSFER Field - ulTargetInfo

ulTargetInfo (ULONG)
Reserved.

Reserved for use by the target. The target can use this field for information about the object and rendering operation.

DRAGTRANSFER Field - usOperation

usOperation (USHORT)
The operation.

Values are:

DO_COPY
Execute a copy operation.

DO_LINK
Execute a link operation.

DO_MOVE
Execute a move operation.

DO_CREATE
Execute a create operation.

DO_NEW
Execute a create another operation. This value should be defined as DO_UNKNOWN+3 if it is not recognized in the current level of the toolkit.

Other
Execute an application-defined operation.

DRAGTRANSFER Field - fsReply

fsReply (USHORT)
Reply flags.

Reply flags for the message. These flags can be set as follows:

DMFL_NATIVERENDER
The source does not support rendering for this object. A source should not set this flag unless it provides sufficient information at the time of the drop for the target to perform the rendering operation. The target must send DM_ENDCONVERSATION to the source after carrying out the rendering operation or when it elects not to do a native rendering.

DMFL_RENDERRETRY
The source supports rendering for the object but does not support the selected rendering mechanism and format. The target can try another mechanism and format by sending another DM_RENDER message. If the target does not retry, it must send a DM_RENDERCOMPLETE message to the source. This flag is set in conjunction with the DMFL_NATIVERENDER flag.

DRIVDATA

Driver-data structure.

```
typedef struct _DRIVDATA {
  LONG     cb;                /*  Length. */
  LONG     lVersion;          /*  Version. */
  CHAR     szDeviceName[32];  /*  Device name. */
  CHAR     abGeneralData[1];  /*  General data. */
} DRIVDATA;

typedef DRIVDATA *PDRIVDATA;
```

DRIVDATA Field - cb

cb (LONG)
Length.

The length of the structure.

DRIVDATA Field - lVersion

lVersion (LONG)
Version.

The version number of the data. Version numbers are defined by particular PM device drivers.

DRIVDATA Field - szDeviceName[32]

szDeviceName[32] (CHAR)
Device name.

A string in a 32-byte field identifying the particular device (model number, and so on). Again, valid values are defined by PM device drivers.

DRIVDATA Field - abGeneralData[1]

abGeneralData[1] (CHAR)
General data.

Data as defined by the Presentation Manager device driver.

The data type of this field is defined by the Presentation Manager device driver. It does not contain pointers, as these are not necessarily valid when passed to the device driver.

EDITCTLINIT

This structure is passed to a details criteria edit window when it is created, so that is knows the current calue and min/max values for the criteria to be edited.

As you modify the compare value using the criteria edit window, you should update the buffer pointed to by pCompareValue.

```
typedef struct _EDITCTLINIT {
  ULONG     cb;                 /*  Length of this structure. */
  PVOID     pCompareValue;      /*  Buffer containing the current value. */
  ULONG     ulLenCompareValue;  /*  Length of the pCompareValue buffer. */
  PVOID     pMinCompareValue;   /*  Set to NULL. */
  PVOID     pMaxCompareValue;   /*  Set to NULL. */
} EDITCTLINIT;

typedef EDITCTLINIT *PEDITCTLINIT;
```

EDITCTLINIT Field - cb

cb (ULONG)
Length of this structure.

EDITCTLINIT Field - pCompareValue

pCompareValue (PVOID)
Buffer containing the current value.

EDITCTLINIT Field - ulLenCompareValue

ulLenCompareValue (ULONG)
Length of the pCompareValue buffer.

EDITCTLINIT Field - pMinCompareValue

pMinCompareValue (PVOID)
Set to NULL.

EDITCTLINIT Field - pMaxCompareValue

pMaxCompareValue (PVOID)

Set to NULL.

FATTRS

Font-attributes structure.

```
typedef struct _FATTRS {
  USHORT     usRecordLength;        /*  Length of record. */
  USHORT     fsSelection;           /*  Selection indicators. */
  LONG       lMatch;                /*  Matched-font identity. */
  CHAR       szFacename[FACESIZE];  /*  Typeface name. */
  USHORT     idRegistry;            /*  Registry identifier. */
  USHORT     usCodePage;            /*  Code page. */
  LONG       lMaxBaselineExt;       /*  Maximum baseline extension. */
  LONG       lAveCharWidth;         /*  Average character width. */
  USHORT     fsType;                /*  Type indicators. */
  USHORT     fsFontUse;             /*  Font-use indicators. */
} FATTRS;

typedef FATTRS *PFATTRS;
```

FATTRS Field - usRecordLength

usRecordLength (USHORT)
Length of record.

FATTRS Field - fsSelection

fsSelection (USHORT)
Selection indicators.

Flags causing the following features to be simulated by the system.

Note: If an italic flag is applied to a font that is itself defined as italic, the font is slanted further by italic simulation.

Underscore or strikeout lines are drawn using the appropriate attributes (for example, color) from the character bundle (see the CHARBUNDLE datatype), not the line bundle (see LINEBUNDLE). The width of the line, and the vertical position of the line in font space, are determined by the font. Horizontally, the line starts from a point in font space directly above or below the start point of each character, and extends to a point directly above or below the escapement point for that character.

For this purpose, the start and escapement points are those applicable to left-to-right or right-to-left character directions (see GpiSetCharDirection in *Graphics Programming Interface Programming Reference*), even if the string is currently being drawn in a top-to-bottom or bottom-to-top direction.

For left-to-right or right-to-left directions, any white space generated by the character extra and character break extra attributes (see GpiSetCharExtra and GpiSetCharBreakExtra in *Graphics Programming Interface Programming Reference*), as well as increments provided by the vector of increments on GpiCharStringPos and GpiCharStringPosAt, are also underlined/overstruck, so that in these cases the line is continuous for the string.

FATTR_SEL_ITALIC	Generate *italic* font.
FATTR_SEL_UNDERSCORE	Generate underscored font.
FATTR_SEL_BOLD	Generate **bold** font. (Note that the resulting characters are wider than those in the original font.)

FATTR_SEL_STRIKEOUT	Generate font with overstruck characters.
FATTR_SEL_OUTLINE	Use an outline font with hollow characters. If this flag is not set, outline font characters are filled. Setting this flag normally gives better performance, and for sufficiently small characters (depending on device resolution) there may be little visual difference.

FATTRS Field - lMatch

lMatch (LONG)
Matched-font identity.

FATTRS Field - szFacename[FACESIZE]

szFacename[FACESIZE] (CHAR)
Typeface name.

The typeface name of the font, for example, Tms Rmn.

FATTRS Field - idRegistry

idRegistry (USHORT)
Registry identifier.

Font registry identifier (zero if unknown).

FATTRS Field - usCodePage

usCodePage (USHORT)
Code page.

If zero, the current Gpi code page (see GpiSetCp in *Graphics Programming Interface Programming Reference*) is used. A subsequent GpiSetCp function changes the code page used for this logical font.

FATTRS Field - lMaxBaselineExt

lMaxBaselineExt (LONG)
Maximum baseline extension.

For raster fonts, this should be the height of the required font, in world coordinates.

For outline fonts, this should be zero.

FATTRS Field - lAveCharWidth

lAveCharWidth (LONG)
Average character width.

For raster fonts, this should be the width of the required font, in world coordinates.

For outline fonts, this should be zero.

FATTRS Field - fsType

fsType (USHORT)
Type indicators.

FATTR_TYPE_KERNING	Enable kerning (PostScript only).
FATTR_TYPE_MBCS	Font for mixed single- and double-byte code pages.
FATTR_TYPE_DBCS	Font for double-byte code pages.
FATTR_TYPE_ANTIALIASED	Antialiased font required. Only valid if supported by the device driver.

FATTRS Field - fsFontUse

fsFontUse (USHORT)
Font-use indicators.

These flags indicate how the font is to be used. They affect presentation speed and font quality.

FATTR_FONTUSE_NOMIX	Text is not mixed with graphics and can be written without regard to any interaction with graphics objects.
FATTR_FONTUSE_OUTLINE	Select an outline (vector) font. The font characters can be used as part of a path definition. If this flag is not set, an outline font might or might not be selected. If an outline font is selected, however, character widths are rounded to an integral number of pels.
FATTR_FONTUSE_TRANSFORMABLE	Characters can be transformed (for example, scaled, rotated, or sheared).

FDATE

Date data structure for file-system functions.

```
typedef struct _FDATE {
  USHORT     day:5;    /*  Binary day for directory entry. */
  USHORT     month:4;  /*  Binary month for directory entry. */
  USHORT     year:7;   /*  The number of years since 1980 for this directory entry. */
} FDATE;

typedef FDATE *PFDATE;
```

FDATE Field - day:5

day:5 (USHORT)
Binary day for directory entry.

FDATE Field - month:4

month:4 (USHORT)
Binary month for directory entry.

FDATE Field - year:7

year:7 (USHORT)
The number of years since 1980 for this directory entry.

FEA2

FEA2 defines the format for setting the full extended attributes in the file.

```
typedef struct _FEA2 {
  ULONG      oNextEntryOffset;  /*  Offset to next entry. */
  BYTE       fEA;               /*  Extended attributes flag. */
  BYTE       cbName;            /*  Length of szName, not including NULL. */
  USHORT     cbValue;           /*  Value length. */
  CHAR       szName[1];         /*  Extended attribute name. */
} FEA2;

typedef FEA2 *PFEA2;
```

Extended attributes (EAs) are non-critical by default. A non-critical EA is one that is not necessary to the functionality of the application. If a non-critical EA is lost, the system continues to operate correctly. For example, losing the icons associated with data files does not generally cause any ill effect other than the inability to show the icon.

A *critical extended attribute* is one which is necessary for the correct operation of the operating system or of a particular operation. EAs

should be marked as critical if their loss would cause the system or program to perform incorrectly. For example, a mail program might store mail headers in EAs. The loss of the header from a message would normally render the mail program unable to further use that message. This would be unacceptable, so the mail program should mark this EA as critical.

FEA2 Field - oNextEntryOffset

oNextEntryOffset (ULONG)
Offset to next entry.

FEA2 Field - fEA

fEA (BYTE)
Extended attributes flag.

FEA_NEEDEA
Extended attributes are critical.

If this flag is set, this file cannot be copied to a file system that does not support extended attributes. This flag should only be set if extended attributes are critical to the processing of this file.

0
Extended attributes are not critical.

FEA2 Field - cbName

cbName (BYTE)
Length of *szName* , not including NULL.

This value must be greater than 0.

FEA2 Field - cbValue

cbValue (USHORT)
Value length.

Sending an EA with *cbValue* set to 0 in the FEA2 data structure causes that attribute to be deleted, if possible. Receiving an EA with *cbValue* set to 0 in the FEA2 data structure indicates that the attribute is not present.

FEA2 Field - szName[1]

szName[1] (CHAR)

Extended attribute name.

FEA2LIST

FEA2 data structure.

```
typedef struct _FEA2LIST {
  ULONG     cbList;   /*  Total bytes of structure including full list. */
  FEA2      list[1];  /*  Variable-length FEA2 structures. */
} FEA2LIST;

typedef FEA2LIST *PFEA2LIST;
```

FEA2LIST Field - cbList

cbList (ULONG)
Total bytes of structure including full list.

FEA2LIST Field - list[1]

list[1] (FEA2)
Variable-length FEA2 structures.

FIELDINFO

Structure that contains information about column data in the details view of the container control. The details view displays each FIELDINFO structure as a column of data that contains specific information about each container record. For example, one FIELDINFO structure, or column, might contain icons or bit maps that represent each container record. Another FIELDINFO structure might contain the date or time that each container record was created.

```
typedef struct _FIELDINFO {
  ULONG                   cb;             /*  Structure size. */
  ULONG                   flData;         /*  Data attributes. */
  ULONG                   flTitle;        /*  Column heading attributes. */
  PVOID                   pTitleData;     /*  Column heading data. */
  ULONG                   offStruct;      /*  Structure offset. */
  PVOID                   pUserData;      /*  Pointer to user data. */
  struct _FIELDINFO      *pNextFieldInfo; /*  Pointer to the next linked FIELDINFO data structure. */
  ULONG                   cxWidth;        /*  Column width. */
} FIELDINFO;

typedef FIELDINFO *PFIELDINFO;
```

FIELDINFO Field - cb

cb (ULONG)
Structure size.

The size (in bytes) of the FIELDINFO structure.

FIELDINFO Field - flData

flData (ULONG)
Data attributes.

Attributes of the data in a field.

- Specify one of the following for each column to choose the type of data that is displayed in each column:

 CFA_BITMAPORICON
 The column contains bit-map or icon data.

 CFA_DATE
 The data in the column is displayed in date format. National Language Support (NLS) is enabled for date format. Use the data structure described in CDATE

 CFA_STRING
 Character or text data is displayed in this column.

 CFA_TIME
 The data in the column is displayed in time format. National Language Support (NLS) is enabled for time format. Use the data structure described in CTIME.

 CFA_ULONG
 Unsigned number data is displayed in this column. National Language Support (NLS) is enabled for number format.

- Specify any or all of the following column attributes:

 CFA_FIREADONLY
 Prevents text in a FIELDINFO data structure (text in a column) from being edited directly. This attribute applies only to columns for which the CFA_STRING attribute has been specified.

 CFA_HORZSEPARATOR
 A horizontal separator is provided beneath column headings.

 CFA_INVISIBLE
 Invisible container column. The default is visible.

 CFA_OWNER
 Ownerdraw is enabled for this container column.

 CFA_SEPARATOR
 A vertical separator is drawn after this column.

- Specify one of the following for each column to vertically position data in that column:

 CFA_BOTTOM
 Bottom-justifies field data.

 CFA_TOP
 Top-justifies field data.

 CFA_VCENTER
 Vertically centers field data. This is the default.

- Specify one of the following for each column to horizontally position data in that column. These attributes can be combined with the attributes used for vertical positioning of column data by using an OR operator (|).

 CFA_CENTER
 Horizontally centers field data.

CFA_LEFT
: Left-justifies field data. This is the default.

CFA_RIGHT
: Right-justifies field data.

FIELDINFO Field - flTitle

flTitle (ULONG)
Column heading attributes.

- Specify the following if icon or bit-map data is to be displayed in the column heading:

 CFA_BITMAPORICON
 : The column heading contains icon or bit-map data. If CFA_BITMAPORICON is not specified, any data that is assigned to a column heading is assumed to be character or text data.

- Specify the following to prevent direct editing of a column heading:

 CFA_FITITLEREADONLY
 : Prevents a column heading from being edited directly.

- Specify one of the following for each column heading to vertically position data in that column heading:

 CFA_TOP
 : Top-justifies column headings.

 CFA_BOTTOM
 : Bottom-justifies column headings.

 CFA_VCENTER
 : Vertically centers column headings. This is the default.

- Specify one of the following for each column heading to horizontally position data in that column heading. These attributes can be combined with the attributes used for vertical positioning of column heading data by using an OR operator (|).

 CFA_CENTER
 : Horizontally centers column headings.

 CFA_LEFT
 : Left-justifies column headings. This is the default.

 CFA_RIGHT
 : Right-justifies column headings.

FIELDINFO Field - pTitleData

pTitleData (PVOID)
Column heading data.

Column heading data, which can be a text string or an icon or bit map. The default is a text string. If the *flTitle* field is set to the CFA_BITMAPORICON attribute, this must be an icon or bit map.

FIELDINFO Field - offStruct

offStruct (ULONG)
Structure offset.

Offset from the beginning of a RECORDCORE structure to the data that is displayed in this column.

Note: If the CCS_MINIRECORDCORE style bit is specified when a container is created, then MINIRECORDCORE should be used instead of RECORDCORE and PMINIRECORDCORE should be used instead of PRECORDCORE in all applicable data structures and messages.

FIELDINFO Field - pUserData

pUserData (PVOID)
Pointer to user data.

FIELDINFO Field - pNextFieldInfo

pNextFieldInfo (struct _FIELDINFO *)
Pointer to the next linked FIELDINFO data structure.

FIELDINFO Field - cxWidth

cxWidth (ULONG)
Column width.

Used to specify the width of a column. The default is an automatically sized column that is always the width of its widest element. If this field is set and the data is too wide, the data is truncated.

FILEFINDBUF4

Level 2 (32-bit) information (used with EAs).

```
typedef struct _FILEFINDBUF4 {
  ULONG     oNextEntryOffset;         /*  Offset of next entry. */
  FDATE     fdateCreation;            /*  Date of file creation. */
  FTIME     ftimeCreation;            /*  Time of file creation. */
  FDATE     fdateLastAccess;          /*  Date of last access. */
  FTIME     ftimeLastAccess;          /*  Time of last access. */
  FDATE     fdateLastWrite;           /*  Date of last write. */
  FTIME     ftimeLastWrite;           /*  Time of last write. */
  ULONG     cbFile;                   /*  Size of file. */
  ULONG     cbFileAlloc;              /*  Allocated size. */
  ULONG     attrFile;                 /*  File attributes. */
  ULONG     cbList;                   /*  Size of the file's extended attributes. */
  UCHAR     cchName;                  /*  Length of file name. */
  CHAR      achName[CCHMAXPATHCOMP];  /*  File name including null terminator. */
} FILEFINDBUF4;

typedef FILEFINDBUF4 *PFILEFINDBUF4;
```

FILEFINDBUF4 Field - oNextEntryOffset

oNextEntryOffset (ULONG)
Offset of next entry.

FILEFINDBUF4 Field - fdateCreation

fdateCreation (FDATE)
Date of file creation.

FILEFINDBUF4 Field - ftimeCreation

ftimeCreation (FTIME)
Time of file creation.

FILEFINDBUF4 Field - fdateLastAccess

fdateLastAccess (FDATE)
Date of last access.

FILEFINDBUF4 Field - ftimeLastAccess

ftimeLastAccess (FTIME)
Time of last access.

FILEFINDBUF4 Field - fdateLastWrite

fdateLastWrite (FDATE)
Date of last write.

FILEFINDBUF4 Field - ftimeLastWrite

ftimeLastWrite (FTIME)
Time of last write.

FILEFINDBUF4 Field - cbFile

cbFile (ULONG)
Size of file.

FILEFINDBUF4 Field - cbFileAlloc

cbFileAlloc (ULONG)
Allocated size.

FILEFINDBUF4 Field - attrFile

attrFile (ULONG)
File attributes.

FILEFINDBUF4 Field - cbList

cbList (ULONG)
Size of the file's extended attributes.

The size is measured in bytes and is the size of the file's entire extended attribute set on the disk.

FILEFINDBUF4 Field - cchName

cchName (UCHAR)
Length of file name.

FILEFINDBUF4 Field - achName[CCHMAXPATHCOMP]

achName[CCHMAXPATHCOMP] (CHAR)
File name including null terminator.

FILTERFLAGS

Filter flags structure.

```
typedef struct _FILTERFLAGS {
  ULONG     cbItem;    /*  Context menu id. */
  ULONG     Flags[3];  /*  Pop-up menu flags. */
} FILTERFLAGS;

typedef FILTERFLAGS *PFILTERFLAGS;
```

FILTERFLAGS Field - cbItem

cbItem (ULONG)
Context menu id.

FILTERFLAGS Field - Flags[3]

Flags[3] (ULONG)
Pop-up menu flags.

FIXED

Signed-integer fraction (16:16). This can be treated as a LONG where the value has been multiplied by 65 536.

```
typedef LONG FIXED;
```

FONTPOT

Fontpot structure.

```
typedef struct _FONTPOT {
  CELL       cell;                  /*  Size of the data that follows. */
  FATTRS     fAttrs;                /*  Font-attribute structure from WinFontDlg. */
  CHAR       szFont[CCHFONTNAME+1]; /*  The font name. */
} FONTPOT;
```

```
typedef FONTPOT *PFONTPOT;
```

FONTPOT Field - cell

cell (CELL)
Size of the data that follows.

FONTPOT Field - fAttrs

fAttrs (FATTRS)
Font-attribute structure from WinFontDlg.

FONTPOT Field - szFont[CCHFONTNAME+1]

szFont[CCHFONTNAME+1] (CHAR)
The font name.

FTIME

Time data structure for file-system functions.

```
typedef struct _FTIME {
  USHORT     twosecs:5;  /*  Binary number of two-second increments. */
  USHORT     minutes:6;  /*  Binary number of minutes. */
  USHORT     hours:5;    /*  Binary number of hours. */
} FTIME;

typedef FTIME *PFTIME;
```

FTIME Field - twosecs:5

twosecs:5 (USHORT)
Binary number of two-second increments.

FTIME Field - minutes:6

minutes:6 (USHORT)
Binary number of minutes.

FTIME Field - hours:5

hours:5 (USHORT)
Binary number of hours.

HAB

Anchor-block handle.

```
typedef LHANDLE HAB;
```

HBITMAP

Bitmap handle.

```
typedef LHANDLE HBITMAP;
```

HCINFO

Hardcopy-capabilities structure.

```
typedef struct _HCINFO {
  CHAR     szFormname[32];  /*  Form name. */
  LONG     cx;              /*  Width (left-to-right) in millimeters. */
  LONG     cy;              /*  Height (top-to-bottom) in millimeters. */
  LONG     xLeftClip;       /*  Left clip limit in millimeters. */
  LONG     yBottomClip;     /*  Bottom clip limit in millimeters. */
  LONG     xRightClip;      /*  Right clip limit in millimeters. */
  LONG     yTopClip;        /*  Top clip limit in millimeters. */
  LONG     xPels;           /*  Number of pels between left and right clip limits. */
  LONG     yPels;           /*  Number of pels between bottom and top clip limits. */
  LONG     flAttributes;    /*  Attributes of the form identifier. */
} HCINFO;

typedef HCINFO *PHCINFO;
```

HCINFO Field - szFormname[32]

szFormname[32] (CHAR)
Form name.

HCINFO Field - cx

cx (LONG)
Width (left-to-right) in millimeters.

HCINFO Field - cy

cy (LONG)
Height (top-to-bottom) in millimeters.

HCINFO Field - xLeftClip

xLeftClip (LONG)
Left clip limit in millimeters.

HCINFO Field - yBottomClip

yBottomClip (LONG)
Bottom clip limit in millimeters.

HCINFO Field - xRightClip

xRightClip (LONG)
Right clip limit in millimeters.

HCINFO Field - yTopClip

yTopClip (LONG)
Top clip limit in millimeters.

HCINFO Field - xPels

xPels (LONG)
Number of pels between left and right clip limits.

HCINFO Field - yPels

yPels (LONG)
Number of pels between bottom and top clip limits.

HCINFO Field - flAttributes

flAttributes (LONG)
Attributes of the form identifier.

HCAPS_SELECTABLE	The form is installed on the printer as given by the printer properties dialog. It is available from an alternate form source without operator intervention. If the form does not have this bit set and is used (if the user selects it), a "forms mismatch" error is generated by the printer object.
HCAPS_CURRENT	The form is the one currently selected by the *pdriv* field (the job properties) in DEVOPENSTRUC

HFIND

Handle associated with a wpclsFindObjectFirst request.

```
typedef LHANDLE HFIND;
```

HMODULE

Module handle.

```
typedef LHANDLE HMODULE;
```

HMQ

Message-queue handle.

```
typedef LHANDLE HMQ;
```

HOBJECT

Workplace object handle.

```
typedef LHANDLE HOBJECT;
```

HPOINTER

Pointer handle.

```
typedef LHANDLE HPOINTER;
```

HPS

Presentation-space handle.

```
typedef LHANDLE HPS;
```

HSTR

String handle.

```
typedef LHANDLE HSTR;
```

HWND

Window handle.

```
typedef LHANDLE HWND;
```

ICONINFO

Icon information data structure.

```
typedef struct _ICONINFO {
  ULONG       cb;           /*  Length of the ICONINFO structure. */
  ULONG       fFormat;      /*  Indicates where the icon resides. */
  PSZ         pszFileName;  /*  Name of the file containing icon data. */
  HMODULE     hmod;         /*  Module containing the icon resource. */
  ULONG       resid;        /*  Identity of the icon resource. */
  ULONG       cbIconData;   /*  Length of the icon data in bytes. */
  PVOID       pIconData;    /*  Pointer to the buffer containing icon data. */
} ICONINFO;

typedef ICONINFO *PICONINFO;
```

ICONINFO Field - cb

cb (ULONG)
Length of the ICONINFO structure.

ICONINFO Field - fFormat

fFormat (ULONG)
Indicates where the icon resides.

Possible values are:

ICON_FILE
Icon file supplied.
ICON_RESOURCE
Icon resource supplied.
ICON_DATA
Icon data supplied.
ICON_CLEAR
Go back to default icon.

ICONINFO Field - pszFileName

pszFileName (PSZ)
Name of the file containing icon data.

This value is ignored if *fFormat* is not equal to ICON_FILE.

ICONINFO Field - hmod

hmod (HMODULE)
Module containing the icon resource.

This value is ignored if *fFormat* is not equal to ICON_RESOURCE.

ICONINFO Field - resid

resid (ULONG)
Identity of the icon resource.

This value is ignored if *fFormat* is not equal to ICON_RESOURCE.

ICONINFO Field - cbIconData

cbIconData (ULONG)
Length of the icon data in bytes.

This value is ignored if *fFormat* is not equal to ICON_DATA.

ICONINFO Field - pIconData

pIconData (PVOID)
Pointer to the buffer containing icon data.

This value is ignored if *fFormat* is not equal to ICON_DATA.

ICONPOS

Icon position structure.

```
typedef struct _ICONPOS {
  POINTL     ptlIcon;        /*  Position of icon. */
  CHAR       szIdentity[1];  /*  Object identity string. */
} ICONPOS;

typedef ICONPOS *PICONPOS;
```

ICONPOS Field - ptlIcon

ptlIcon (POINTL)
Position of icon.

ICONPOS Field - szIdentity[1]

szIdentity[1] (CHAR)
Object identity string.

LHANDLE

The handle of a resource.

```
typedef unsigned long LHANDLE;
```

LINEBUNDLE

Line-attributes bundle structure.

```
typedef struct _LINEBUNDLE {
  LONG       lColor;          /*  Line foreground color. */
  LONG       lBackColor;      /*  Line background color. */
  USHORT     usMixMode;       /*  Line foreground-mix mode. */
  USHORT     usBackMixMode;   /*  Line background-mix mode. */
  FIXED      fxWidth;         /*  Line width. */
  LONG       lGeomWidth;      /*  Geometric line width. */
  USHORT     usType;          /*  Line type. */
  USHORT     usEnd;           /*  Line end. */
  USHORT     usJoin;          /*  Line join. */
  USHORT     usReserved;      /*  Reserved. */
} LINEBUNDLE;

typedef LINEBUNDLE *PLINEBUNDLE;
```

LINEBUNDLE Field - lColor

lColor (LONG)
Line foreground color.

LINEBUNDLE Field - lBackColor

lBackColor (LONG)
Line background color.

LINEBUNDLE Field - usMixMode

usMixMode (USHORT)
Line foreground-mix mode.

LINEBUNDLE Field - usBackMixMode

usBackMixMode (USHORT)
Line background-mix mode.

LINEBUNDLE Field - fxWidth

fxWidth (FIXED)
Line width.

LINEBUNDLE Field - lGeomWidth

lGeomWidth (LONG)
Geometric line width.

LINEBUNDLE Field - usType

usType (USHORT)
Line type.

LINEBUNDLE Field - usEnd

usEnd (USHORT)
Line end.

LINEBUNDLE Field - usJoin

usJoin (USHORT)
Line join.

LINEBUNDLE Field - usReserved

usReserved (USHORT)
Reserved.

LINKITEM

USAGE_LINK structure.

```
typedef struct _LINKITEM {
  WPObject      *LinkObj;  /*  The link object. */
} LINKITEM;

typedef LINKITEM *PLINKITEM;
```

LINKITEM Field - LinkObj

LinkObj (WPObject *)
The link object.

Points to an object of class WPObject.

LONG

Signed integer in the range -2 147 483 648 through 2 147 483 647.

```
#define LONG long
```

Note: Where this data type represents a graphic coordinate in world or model space, its value is restricted to -134 217 728 through 134 217 727.

A graphic coordinate in device or screen coordinates is restricted to -32 768 through 32 767.

The value of a graphic coordinate may be further restricted by any transforms currently in force, including the positioning of the origin of the window on the screen. In particular, coordinates in world or model space must not generate coordinate values after transformation (that is, in device or screen space) outside the range -32 768 through 32 767.

MEMORYITEM

USAGE_MEMORY structure.

```
typedef struct _MEMORYITEM {
  ULONG     cbBuffer;  /*  Number of bytes in the memory block. */
} MEMORYITEM;

typedef MEMORYITEM *PMEMORYITEM;
```

MEMORYITEM Field - cbBuffer

cbBuffer (ULONG)
Number of bytes in the memory block.

MINIRECORDCORE

Structure that contains information for records smaller than those defined by the RECORDCORE data structure. This data structure is used if the CCS_MINIRECORDCORE style bit is specified when a container is created.

```
typedef struct _MINIRECORDCORE {
  ULONG                         cb;               /*  Structure size. */
  ULONG                         flRecordAttr;     /*  Attributes of container records. */
  POINTL                        ptlIcon;          /*  Record position. */
  struct _MINIRECORDCORE       *preccNextRecord;  /*  Pointer to the next linked record. */
  PSZ                           pszIcon;          /*  Record text. */
  HPOINTER                      hptrIcon;         /*  Record icon. */
} MINIRECORDCORE;

typedef MINIRECORDCORE *PMINIRECORDCORE;
```

MINIRECORDCORE Field - cb

cb (ULONG)
Structure size.

The size (in bytes) of the MINIRECORDCORE structure.

MINIRECORDCORE Field - flRecordAttr

flRecordAttr (ULONG)
Attributes of container records.

Contains any or all of the following:

CRA_COLLAPSED
Specifies that a record is collapsed.

CRA_CURSORED
Specifies that a record will be drawn with a selection cursor.

CRA_DROPONABLE
Specifies that a record can be a target for direct manipulation.

CRA_EXPANDED
Specifies that a record is expanded.

CRA_FILTERED
Specifies that a record is filtered and, therefore, hidden from view.

CRA_INUSE
Specifies that a record will be drawn with in-use emphasis.

CRA_RECORDREADONLY
Prevents a record from being edited directly.

CRA_SELECTED
Specifies that a record will be drawn with selected-state emphasis.

CRA_TARGET
Specifies that a record will be drawn with target emphasis.

MINIRECORDCORE Field - ptlIcon

ptlIcon (POINTL)
Record position.

Position of a container record in the icon view.

MINIRECORDCORE Field - preccNextRecord

preccNextRecord (struct _MINIRECORDCORE *)
Pointer to the next linked record.

MINIRECORDCORE Field - pszIcon

pszIcon (PSZ)
Record text.

Text for the container record.

MINIRECORDCORE Field - hptrIcon

hptrIcon (HPOINTER)
Record icon.

Icon that is displayed for the container record.

MPARAM

A 4-byte message-dependent parameter structure.

```
typedef VOID *MPARAM;
```

Certain elements of information, placed into the parameters of a message, have data types that do not use all four bytes of this data type. The rules governing these cases are:

BOOL	The value is contained in the low word and the high word is 0.
SHORT	The value is contained in the low word and its sign is extended into the high word.
USHORT	The value is contained in the low word and the high word is 0.
NULL	The entire four bytes are 0.

The structure of this data type depends on the message. For details, see the description of the particular message.

MRESULT

A 4-byte message-dependent reply parameter structure.

```
typedef VOID *MRESULT;
```

Certain elements of information, placed into the parameters of a message, have data types that do not use all four bytes of this data type. The rules governing these cases are:

BOOL	The value is contained in the low word and the high word is 0.
SHORT	The value is contained in the low word and its sign is extended into the high word.
USHORT	The value is contained in the low word and the high word is 0.
NULL	The entire four bytes are 0.

The structure of this data type depends on the message. For details, see the description of the particular message.

OBJCLASS

Object class structure.

```
typedef struct _OBJCLASS {
  struct _OBJCLASS      *pNext;         /*  Pointer to the next object class structure. */
```

```
  PSZ                   pszClassName;  /*  Class name. */
  PSZ                   pszModName;    /*  Module name. */
} OBJCLASS;

typedef OBJCLASS *POBJCLASS;
```

OBJCLASS Field - pNext

pNext (struct _OBJCLASS *)
Pointer to the next object class structure.

OBJCLASS Field - pszClassName

pszClassName (PSZ)
Class name.

OBJCLASS Field - pszModName

pszModName (PSZ)
Module name.

OBJDATA

Object data structure. Class-specific information is contained in this structure.

```
typedef struct _OBJDATA {
  WPSRCLASSBLOCK      *CurrentClass;  /*  Pointer to the current save or restore class block. */
  WPSRCLASSBLOCK      *First;         /*  Pointer to the first save or restore class block. */
  PUCHAR               NextData;      /*  Pointer to the next block of data. */
  USHORT               Length;        /*  Length. */
} OBJDATA;

typedef OBJDATA *POBJDATA;
```

OBJDATA Field - CurrentClass

CurrentClass (WPSRCLASSBLOCK *)
Pointer to the current save or restore class block.

OBJDATA Field - First

First (WPSRCLASSBLOCK *)
Pointer to the first save or restore class block.

OBJDATA Field - NextData

NextData (PUCHAR)
Pointer to the next block of data.

OBJDATA Field - Length

Length (USHORT)
Length.

OBJECT

Pointer to an object of class WPObject.

```
typedef WPObject *OBJECT;
```

OWNERITEM

Owner item.

```
typedef struct _OWNERITEM {
  HWND      hwnd;            /*  Window handle. */
  HPS       hps;             /*  Presentation-space handle. */
  ULONG     fsState;         /*  State. */
  ULONG     fsAttribute;     /*  Attribute. */
  ULONG     fsStateOld;      /*  Old state. */
  ULONG     fsAttributeOld;  /*  Old attribute. */
  RECTL     rclItem;         /*  Item rectangle. */
  LONG      idItem;          /*  Item identity. */
  ULONG     hItem;           /*  Item. */
} OWNERITEM;

typedef OWNERITEM *POWNERITEM;
```

OWNERITEM Field - hwnd

hwnd (HWND)
Window handle.

OWNERITEM Field - hps

hps (HPS)
Presentation-space handle.

OWNERITEM Field - fsState

fsState (ULONG)
State.

OWNERITEM Field - fsAttribute

fsAttribute (ULONG)
Attribute.

OWNERITEM Field - fsStateOld

fsStateOld (ULONG)
Old state.

OWNERITEM Field - fsAttributeOld

fsAttributeOld (ULONG)
Old attribute.

OWNERITEM Field - rclItem

rclItem (RECTL)
Item rectangle.

OWNERITEM Field - idItem

idItem (LONG)
Item identity.

OWNERITEM Field - hItem

hItem (ULONG)
Item.

PAGEINFO

Settings page information structure.

```
typedef struct _PAGEINFO {
  ULONG       cb;                 /*  Length of PAGEINFO structure. */
  HWND        hwndPage;           /*  Handle of page. */
  PFNWP       pfnwp;              /*  Window procedure. */
  ULONG       resid;              /*  Resource identity. */
  PVOID       pCreateParams;      /*  Pointer to creation parameters. */
  USHORT      dlgid;              /*  Dialog identity. */
  USHORT      usPageStyleFlags;   /*  Notebook control-page style flags. */
  USHORT      usPageInsertFlags;  /*  Notebook control-page insertion flags. */
  USHORT      usSettingsFlags;    /*  Settings flag. */
  PSZ         pszName;            /*  Pointer to a string containing the page name. */
  USHORT      idDefaultHelpPanel; /*  Identity of the default help panel. */
  USHORT      usReserved2;        /*  Reserved value; must be zero. */
  PSZ         pszHelpLibraryName; /*  Pointer to the name of the help file. */
  PUSHORT     pHelpSubtable;      /*  Pointer to the help subtable. */
  HMODULE     hmodHelpSubtable;   /*  Module handle for the help subtable. */
  ULONG       ulPageInsertId;     /*  Notebook control-page identity. */
} PAGEINFO;

typedef PAGEINFO *PPAGEINFO;
```

PAGEINFO Field - cb

cb (ULONG)
Length of PAGEINFO structure.

PAGEINFO Field - hwndPage

hwndPage (HWND)
Handle of page.

PAGEINFO Field - pfnwp

pfnwp (PFNWP)
Window procedure.

PAGEINFO Field - resid

resid (ULONG)
Resource identity.

PAGEINFO Field - pCreateParams

pCreateParams (PVOID)
Pointer to creation parameters.

PAGEINFO Field - dlgid

dlgid (USHORT)
Dialog identity.

PAGEINFO Field - usPageStyleFlags

usPageStyleFlags (USHORT)
Notebook control-page style flags.

PAGEINFO Field - usPageInsertFlags

usPageInsertFlags (USHORT)
Notebook control-page insertion flags.

PAGEINFO Field - usSettingsFlags

usSettingsFlags (USHORT)
Settings flag.

This flag must be set to one of the following values:

0
You will not get page numbers.

SETTINGS_PAGE_NUMBERS
Page numbers will automatically be put on the status line for pages that have minor pages under the major tab page.

If you want to use the page numbers, make sure all pages have this setting.

PAGEINFO Field - pszName

pszName (PSZ)
Pointer to a string containing the page name.

PAGEINFO Field - idDefaultHelpPanel

idDefaultHelpPanel (USHORT)
Identity of the default help panel.

PAGEINFO Field - usReserved2

usReserved2 (USHORT)
Reserved value; must be zero.

PAGEINFO Field - pszHelpLibraryName

pszHelpLibraryName (PSZ)
Pointer to the name of the help file.

PAGEINFO Field - pHelpSubtable

pHelpSubtable (PUSHORT)
Pointer to the help subtable.

PAGEINFO Field - hmodHelpSubtable

hmodHelpSubtable (HMODULE)
Module handle for the help subtable.

PAGEINFO Field - ulPageInsertId

ulPageInsertId (ULONG)
Notebook control-page identity.

PAINTPOT

PAINTPOT structure.

```
typedef struct _PAINTPOT {
  CELL      cell;   /*  Size of the data that follows. */
  ULONG     ulRGB;  /*  Color of this paintpot. */
} PAINTPOT;

typedef PAINTPOT *PPAINTPOT;
```

PAINTPOT Field - cell

cell (CELL)
Size of the data that follows.

PAINTPOT Field - ulRGB

ulRGB (ULONG)
Color of this paintpot.

PALINFO

Class-specific palette information data.

```
typedef struct _PALINFO {
  ULONG     xCellCount;   /*  Number of columns of palinfos. */
  ULONG     yCellCount;   /*  Number of rows of palinfos. */
  ULONG     xCursor;      /*  Cursor location (read only). */
  ULONG     yCursor;      /*  Cursor location (read only). */
  ULONG     xCellWidth;   /*  Width of each palinfo. */
  ULONG     yCellHeight;  /*  Height of each palinfo. */
  ULONG     xGap;         /*  X separation of palinfos. */
  ULONG     yGap;         /*  Y separation of palinfos. */
} PALINFO;

typedef PALINFO *PPALINFO;
```

PALINFO Field - xCellCount

xCellCount (ULONG)
Number of columns of palinfos.

PALINFO Field - yCellCount

yCellCount (ULONG)
Number of rows of palinfos.

PALINFO Field - xCursor

xCursor (ULONG)
Cursor location (read only).

PALINFO Field - yCursor

yCursor (ULONG)
Cursor location (read only).

PALINFO Field - xCellWidth

xCellWidth (ULONG)
Width of each palinfo.

PALINFO Field - yCellHeight

yCellHeight (ULONG)
Height of each palinfo.

PALINFO Field - xGap

xGap (ULONG)
X separation of palinfos.

PALINFO Field - yGap

yGap (ULONG)
Y separation of palinfos.

PCSZ

Pointer to a constant null-terminated string.

```
typedef const char *PCSZ;
```

PDEVOPENDATA

Open device-data array.

This data type points to data whose format is described by the DEVOPENSTRUC data type.

```
typedef PSZ *PDEVOPENDATA;
```

PFNCOMPARE

Pointer to a Details View comparison function.

This function would be used if you wanted to sort on a non-standard data type or if you wanted to a custom sort.

```
typedef FNCOMPARE *PFNCOMPARE;
```

In the header file, this is a two-part definition as shown below:

```
typedef LONG (EXPENTRY FNCOMPARE)(PVOID, PVOID);
typedef FNCOMPARE *PFNCOMPARE;
```

The first argument (PVOID) is a pointer to the Details View field value. The second argument (PVOID) is a pointer to the value it is to be compared with. The function returns a LONG value that must be one of the following:

CMP_EQUAL	0
CMP_GREATER	1
CMP_LESS	2

PFNOWNDRW

Pointer to an owner-draw function for a Details View field.

This function would be used if you wanted to draw the contents of the Details View field yourself, for example, as a graphic image or to use custom formatting.

This function is called each time the field is painted.

```
typedef FNOWNDRW *PFNOWNDRW;
```

The first argument (HWND) is the handle of the Details View field-window being painted. The second argument (PVOID) is a pointer to an OWNERITEM data structure. The third argument (PVOID) is a pointer to the Comparison value to draw (this argument is NULL except when this function is painting its value into the Include Criteria Dialog (in the Settings notebook), in which case it points to a string containing a comparision value. The function returns a BOOL value of TRUE if the field was owner-drawn or FALSE if the container control should do default painting.

In the header file, this is a two-part definition as shown below:

```
typedef BOOL (EXPENTRY FNOWNDRW)(HWND, PVOID, PVOID);
typedef FNOWNDRW *PFNOWNDRW;
```

PFNWP

Pointer to a window procedure.

This is the standard function definition for window procedures.

```
typedef FNWP *PFNWP;
```

The first argument (HWND) is the handle of the window receiving the message. The second argument (ULONG) is a message identifier. The third argument (MPARAM) is the first message parameter (mp1). The fourth argument (MPARAM) is the second message parameter (mp2). The function returns an MRESULT. Each message has a specific set of possible return codes. The window procedure must return a value that

is appropriate for the message being processed.

In the header file, this is a two-part definition as shown below:

```
typedef MRESULT (EXPENTRY FNWP)(HWND, ULONG, MPARAM, MPARAM);
typedef FNWP *PFNWP;
```

Window procedures must be EXPORTED in the definitions file used by the linker.

POINTL

Point structure (long integers).

```
typedef struct _POINTL {
  LONG     x;  /*  X-coordinate. */
  LONG     y;  /*  Y-coordinate. */
} POINTL;

typedef POINTL *PPOINTL;
```

POINTL Field - x

x (LONG)
X-coordinate.

POINTL Field - y

y (LONG)
Y-coordinate.

PRINTDEST

PRINTDEST data structure.

Contains all the parameters required to issue DevPostDeviceModes and DevOpenDC function calls.

```
typedef struct _PRINTDEST {
  ULONG            cb;          /*  Length of data structure, in bytes. */
  LONG             lType;       /*  Type of device context. */
  PSZ              pszToken;    /*  Device-information token. */
  LONG             lCount;      /*  Number of items. */
  PDEVOPENDATA     pdopData;    /*  Open device context data area. */
  ULONG            fl;          /*  Flags. */
  PSZ              pszPrinter;  /*  Name of the printer. */
} PRINTDEST;

typedef PRINTDEST *PPRINTDEST;
```

PRINTDEST Field - cb

cb (ULONG)
Length of data structure, in bytes.

PRINTDEST Field - lType

lType (LONG)
Type of device context.

OD_QUEUED
The device context is queued.
OD_DIRECT
The device context is direct.

PRINTDEST Field - pszToken

pszToken (PSZ)
Device-information token.

This is always "*".

PRINTDEST Field - lCount

lCount (LONG)
Number of items.

This is the number of items present in the *pdopData* field.

PRINTDEST Field - pdopData

pdopData (PDEVOPENDATA)
Open device context data area.

See DEVOPENSTRUC for information on the format of *pdopData* .

PRINTDEST Field - fl

fl (ULONG)
Flags.

PD_JOB_PROPERTY
This flag indicates that DevPostDeviceModes should be called with DPDM_POSTJOBPROP before calling DevOpenDC.

PRINTDEST Field - pszPrinter

pszPrinter (PSZ)
Name of the printer.

A name that specifies the device; for example, "PRINTER1". The name is used for calling DevPostDeviceModes.

The printer device name can be found by calling SplQueryQueue and passing to it the information found in the *pszLogAddress* field of the DEVOPENSTRUC structure pointed to by *pdopData*. SplQueryQueue returns a PRQINFO3 structure. The *pszPrinters* field in PRQINFO3 contains the printer device name to be used.

PROGCATEGORY

Program category.

```
typedef ULONG PROGCATEGORY;
```

PROGDETAILS

Program-details structure.

```
typedef struct _PROGDETAILS {
  ULONG        Length;          /*  Length of structure. */
  PROGTYPE     progt;           /*  Program type. */
  PSZ          pszTitle;        /*  Title. */
  PSZ          pszExecutable;   /*  Executable file name (program name). */
  PSZ          pszParameters;   /*  Parameter string. */
  PSZ          pszStartupDir;   /*  Start-up directory. */
  PSZ          pszIcon;         /*  Icon-file name. */
  PSZ          pszEnvironment;  /*  Environment string. */
  SWP          swpInitial;      /*  Initial window position and size. */
} PROGDETAILS;

typedef PROGDETAILS *PPROGDETAILS;
```

PROGDETAILS Field - Length

Length (ULONG)

Length of structure.

PROGDETAILS Field - progt

progt (PROGTYPE)
Program type.

PROGDETAILS Field - pszTitle

pszTitle (PSZ)
Title.

If pszTitle is NULL, pszExecutable is used.

PROGDETAILS Field - pszExecutable

pszExecutable (PSZ)
Executable file name (program name).

If pszExecutable is NULL, the file specified in the OS2_SHELL statement in the configuration file (CONFIG.SYS) is used. If the OS2_SHELL statement does not exist, the command processor (CMD.EXE for a non-DOS session, or COMMAND.COM for a DOS session) is used.

PROGDETAILS Field - pszParameters

pszParameters (PSZ)
Parameter string.

The pszParameters can be set to NULL.

PROGDETAILS Field - pszStartupDir

pszStartupDir (PSZ)
Start-up directory.

The pszStartupDir can be set to NULL.

PROGDETAILS Field - pszIcon

pszIcon (PSZ)
Icon-file name.

If pszIcon is NULL, the default icon for the executable is used.

PROGDETAILS Field - pszEnvironment

pszEnvironment (PSZ)
Environment string.

A list of null-terminated strings ending with an extra NULL character.

PROGDETAILS Field - swpInitial

swpInitial (SWP)
Initial window position and size.

PROGTYPE

Program-type structure.

```
typedef struct _PROGTYPE {
  PROGCATEGORY     progc;       /*  Program category: */
  ULONG            fbVisible;   /*  Visibility attribute. */
} PROGTYPE;

typedef PROGTYPE *PPROGTYPE;
```

PROGTYPE Field - progc

progc (PROGCATEGORY)
Program category:

PROG_DEFAULT
: Default application.

PROG_PM
: Presentation Manager application.

PROG_WINDOWABLEVIO
: Text-windowed application.

PROG_FULLSCREEN
: Full-screen application.

PROG_WINDOWEDVDM
: PC DOS executable process (windowed).

PROG_VDM
: PC DOS executable process (full screen).

PROG_REAL
: PC DOS executable process (full screen). Same as PROG_VDM.

PROG_31_STDSEAMLESSVDM
: Windows 3.1 program that will execute in its own windowed WINOS2 session.

PROG_31_STDSEAMLESSCOMMON
: Windows 3.1 program that will execute in a common windowed WINOS2 session.

PROG_31_ENHSEAMLESSVDM
: Windows 3.1 program that will execute in enhanced compatibility mode in its own windowed WINOS2 session.

PROG_31_ENHSEAMLESSCOMMON
: Windows 3.1 program that will execute in enhanced compatibility mode in a common windowed WINOS2 session.

PROG_31_ENH
: Windows 3.1 program that will execute in enhanced compatibility mode in a full-screen WINOS2 session.

PROG_31_STD
: Windows 3.1 program that will execute in a full-screen WINOS2 session.

PROGTYPE Field - fbVisible

fbVisible (ULONG)
: Visibility attribute.

When testing this field, allow for the possibility that other bits may be defined in the future. SHE_INVISIBLE and SHE_PROTECTED can be used to mask the visibility and protected flags, respectively.

SHE_VISIBLE
: Visible

SHE_INVISIBLE
: Invisible

SHE_UNPROTECTED
: Unprotected

SHE_PROTECTED
: Protected.

PRQINFO3

Print-queue information structure.

This structure is used at information levels 3 and 4.

```
typedef struct _PRQINFO3 {
  PSZ           pszName;        /*  Queue name. */
  USHORT        uPriority;      /*  Queue priority. */
  USHORT        uStartTime;     /*  Minutes after midnight when queue becomes active. */
  USHORT        uUntilTime;     /*  Minutes after midnight. when queue ceases to be active. */
  USHORT        fsType;         /*  Queue type. */
  PSZ           pszSepFile;     /*  Separator-page file. */
  PSZ           pszPrProc;      /*  Default queue-processor. */
  PSZ           pszParms;       /*  Queue parameters. */
  PSZ           pszComment;     /*  Queue description. */
  USHORT        fsStatus;       /*  Queue status. */
  USHORT        cJobs;          /*  Number of jobs in queue. */
  PSZ           pszPrinters;    /*  Print devices connected to queue. */
  PSZ           pszDriverName;  /*  Default device driver. */
  PDRIVDATA     pDriverData;    /*  Default queue job properties. */
} PRQINFO3;

typedef PRQINFO3 *PPRQINFO3;
```

PRQINFO3 Field - pszName

pszName (PSZ)
Queue name.

The maximum length of the name in the network case is 256 (including one byte for zero termination).

PRQINFO3 Field - uPriority

uPriority (USHORT)
Queue priority.

The range is 1 through 9, with 1 being the highest queue priority.

The default job priority (DefJobPrio) is determined from:
DefJobPrio=100-(10* *uPriority*).

If a job is added with **PRJ_NO_PRIORITY** specified, DefJobPrio is used. If a default priority higher than the default job priority is specified, the default job priority is used. If a default priority lower than the default is specified, the specified job priority is used.

PRQ_DEF_PRIORITY
Default priority
PRQ_MAX_PRIORITY
Highest priority
PRQ_MIN_PRIORITY
Minimum priority
PRQ_NO_PRIORITY
No priority.

PRQINFO3 Field - uStartTime

uStartTime (USHORT)
Minutes after midnight when queue becomes active.

For example, the value 75 represents 1:15 a.m.

If *uStartTime* and *uUntilTime* are both 0, the print queue is always available.

PRQINFO3 Field - uUntilTime

uUntilTime (USHORT)
Minutes after midnight. when queue ceases to be active.

For example, the value 1200 represents 8 p.m.

If *uUntilTime* and *uStartTime* are both 0, the print queue is always available.

PRQINFO3 Field - fsType

fsType (USHORT)
Queue type.

PRQ3_TYPE_RAW
Data is always enqueued in the device specific format.
PRQ3_TYPE_BYPASS
Allows the spooler to bypass the queue processor and send data directly to the Printer Driver. Setting this bit allows the spooler to print jobs of type PM_Q_RAW while they are still being spooled.
PRQ3_TYPE_APPDEFAULT
This bit is set for the application default queue only.

PRQINFO3 Field - pszSepFile

pszSepFile (PSZ)
Separator-page file.

The path and file name of a separator-page file on the target computer.

This file contains formatting information for the page or pages to be used between print jobs. A relative path name is taken as relative to the current spool directory. A NULL string indicates no separator page.

PRQINFO3 Field - pszPrProc

pszPrProc (PSZ)
Default queue-processor.

PRQINFO3 Field - pszParms

pszParms (PSZ)
Queue parameters.

This can be any text string or a NULL string.

PRQINFO3 Field - pszComment

pszComment (PSZ)
Queue description.

A NULL string results in no comment. The maximum length is 48 characters (including one byte for the null terminator).

PRQINFO3 Field - fsStatus

fsStatus (USHORT)
Queue status.

PRQ3_PAUSED
Queue is paused (held).
PRQ3_PENDING
Queue is pending deletion.

PRQINFO3 Field - cJobs

cJobs (USHORT)
Number of jobs in queue.

PRQINFO3 Field - pszPrinters

pszPrinters (PSZ)
Print devices connected to queue.

This cannot be NULL.

PRQINFO3 Field - pszDriverName

pszDriverName (PSZ)
Default device driver.

PRQINFO3 Field - pDriverData

pDriverData (PDRIVDATA)
Default queue job properties.

Note: An application can use *pszDriverName* , *pDriverData* , *pszPrProc* , and *pszParms* to construct a valid DevOpenDC call based only on the queue name.

PSZ

Pointer to a null-terminated string.

If you are using C++ **, you may need to use PCSZ.

```
typedef unsigned char *PSZ;
```

PVOID

Pointer to a data type of undefined format.

```
typedef VOID *PVOID;
```

RECORDCORE

Structure that contains information for records in a container control. This data structure is used if the CCS_MINIRECORDCORE style bit is not specified when a container is created.

```
typedef struct _RECORDCORE {
  ULONG                     cb;               /*  The size, in bytes, of the RECORDCORE structure. */
  ULONG                     flRecordAttr;     /*  Container record attributes. */
  POINTL                    ptlIcon;          /*  Position of a container record in the icon view. */
  struct _RECORDCORE       *preccNextRecord;  /*  Pointer to the next linked record. */
  PSZ                       pszIcon;          /*  Text for the icon view (CV_ICON). */
  HPOINTER                  hptrIcon;         /*  Icon that is displayed when the CV_MINI style bit is not specif
  HPOINTER                  hptrMiniIcon;     /*  Icon that is displayed when the CV_MINI style bit is specified.
  HBITMAP                   hbmBitmap;        /*  Bit map displayed when the CV_MINI style bit is not specified.
  HBITMAP                   hbmMiniBitmap;    /*  Bit map displayed when the CV_MINI style bit is specified. */
  PTREEITEMDESC             pTreeItemDesc;    /*  Pointer to a TREEITEMDESC structure. */
  PSZ                       pszText;          /*  Text for the text view (CV_TEXT). */
  PSZ                       pszName;          /*  Text for the name view (CV_NAME). */
  PSZ                       pszTree;          /*  Text for the tree view (CV_TREE). */
} RECORDCORE;

typedef RECORDCORE *PRECORDCORE;
```

RECORDCORE Field - cb

cb (ULONG)
The size, in bytes, of the RECORDCORE structure.

RECORDCORE Field - flRecordAttr

flRecordAttr (ULONG)
Container record attributes.

This parameter can contain any or all of the following:

CRA_COLLAPSED
Specifies that a record is collapsed.

CRA_CURSORED
Specifies that a record will be drawn with a selection cursor.

CRA_DISABLED
Specifies that a record will be drawn with unavailable-state emphasis.

CRA_DROPONABLE
Specifies that a record can be a target for direct manipulation.

CRA_EXPANDED
Specifies that a record is expanded.

CRA_FILTERED
Specifies that a record is filtered and, therefore, hidden from view.

CRA_INUSE
Specifies that a record will be drawn with in-use emphasis.

CRA_PICKED
Specifies that the container record willl be picked up as part of the drag set.

CRA_SELECTED
Specifies that a record will be drawn with selected-state emphasis.

CRA_SOURCE
Specifies that a record will be drawn with source-menu emphasis.

RECORDCORE Field - ptlIcon

ptlIcon (POINTL)
Position of a container record in the icon view.

RECORDCORE Field - preccNextRecord

preccNextRecord (struct _RECORDCORE *)
Pointer to the next linked record.

RECORDCORE Field - pszIcon

pszIcon (PSZ)
Text for the icon view (CV_ICON).

RECORDCORE Field - hptrIcon

hptrIcon (HPOINTER)
Icon that is displayed when the CV_MINI style bit is not specified.

This field is used when the CA_DRAWICON container attribute of the CNRINFO data structure is set.

RECORDCORE Field - hptrMiniIcon

hptrMiniIcon (HPOINTER)
Icon that is displayed when the CV_MINI style bit is specified.

This field is used when the CA_DRAWICON container attribute of the CNRINFO data structure is set.

RECORDCORE Field - hbmBitmap

hbmBitmap (HBITMAP)
Bit map displayed when the CV_MINI style bit is not specified.

This field is used when the CA_DRAWBITMAP container attribute of the CNRINFO data structure is set.

RECORDCORE Field - hbmMiniBitmap

hbmMiniBitmap (HBITMAP)
Bit map displayed when the CV_MINI style bit is specified.

This field is used when the CA_DRAWBITMAP container attribute of the CNRINFO data structure is set.

RECORDCORE Field - pTreeItemDesc

pTreeItemDesc (PTREEITEMDESC)
Pointer to a TREEITEMDESC structure.

The TREEITEMDESC structure contains the icons and bit maps used to represent the state of an expanded or collapsed parent item in the tree name view.

RECORDCORE Field - pszText

pszText (PSZ)
Text for the text view (CV_TEXT).

RECORDCORE Field - pszName

pszName (PSZ)
Text for the name view (CV_NAME).

RECORDCORE Field - pszTree

pszTree (PSZ)
Text for the tree view (CV_TREE).

RECORDINSERT

Structure that contains information about RECORDCORE structures that are being inserted into a container. The RECORDINSERT structure is used in the CM_INSERTRECORD container message only.

Note: If the CCS_MINIRECORDCORE style bit is specified when a container is created, then MINIRECORDCORE should be used instead of RECORDCORE and PMINIRECORDCORE should be used instead of PRECORDCORE in all applicable data structures and messages.

```
typedef struct _RECORDINSERT {
  ULONG           cb;                 /*  Structure size. */
  PRECORDCORE     pRecordOrder;       /*  Record order. */
  PRECORDCORE     pRecordParent;      /*  Pointer to record parent. */
  ULONG           fInvalidateRecord;  /*  Update flag. */
  ULONG           zOrder;             /*  Record z-order. */
  ULONG           cRecordsInsert;     /*  Number of root-level structures. */
} RECORDINSERT;

typedef RECORDINSERT *PRECORDINSERT;
```

RECORDINSERT Field - cb

cb (ULONG)
Structure size.

The size (in bytes) of the RECORDINSERT structure.

RECORDINSERT Field - pRecordOrder

pRecordOrder (PRECORDCORE)

Record order.

Orders the RECORDCORE structures relative to other RECORDCORE structures in the container. The values can be:

CMA_FIRST
: Places a RECORDCORE structure, or list of RECORDCORE structures, at the beginning of the list of structures.

CMA_END
: Places a RECORDCORE structure, or list of RECORDCORE structures, at the end of the list of structures.

Other
: Pointer to a RECORDCORE structure after which this structure, or list of structures, is to be inserted.

RECORDINSERT Field - pRecordParent

pRecordParent (PRECORDCORE)
Pointer to record parent.

Pointer to a RECORDCORE structure that is the parent of the records to be inserted. This field is used only with the CMA_FIRST or CMA_END attributes of the *pRecordOrder* field.

RECORDINSERT Field - fInvalidateRecord

fInvalidateRecord (ULONG)
Update flag.

Flag that indicates an automatic display update after RECORDCORE structures are inserted.

TRUE
: The display is automatically updated after a RECORDCORE structure is inserted.

FALSE
: The application must send the CM_INVALIDATERECORD message after a RECORDCORE structure is inserted.

RECORDINSERT Field - zOrder

zOrder (ULONG)
Record z-order.

Positions the RECORDCORE structure in z-order, relative to other records in the container. The values can be:

CMA_TOP
: Places a RECORDCORE structure at the top of the z-order. This is the default value.

CMA_BOTTOM
: Places a RECORDCORE structure at the bottom of the z-order.

RECORDINSERT Field - cRecordsInsert

cRecordsInsert (ULONG)
Number of root-level structures.

The number of root-level RECORDCORE structures to be inserted. The *cRecordsInsert* field value must be greater than 0.

RECORDITEM

USAGE_RECORD structure.

```
typedef struct _RECORDITEM {
  HWND                hwndCnr;  /*  Container into which the object is inserted. */
  PMINIRECORDCORE     pRecord;  /*  Record pointer within the container. */
  ULONG               ulUser;   /*  For application use. */
} RECORDITEM;

typedef RECORDITEM *PRECORDITEM;
```

RECORDITEM Field - hwndCnr

hwndCnr (HWND)
Container into which the object is inserted.

RECORDITEM Field - pRecord

pRecord (PMINIRECORDCORE)
Record pointer within the container.

RECORDITEM Field - ulUser

ulUser (ULONG)
For application use.

RECTL

Rectangle structure.

```
typedef struct _RECTL {
  LONG     xLeft;    /*  X-coordinate of left-hand edge of rectangle. */
  LONG     yBottom;  /*  Y-coordinate of bottom edge of rectangle. */
  LONG     xRight;   /*  X-coordinate of right-hand edge of rectangle. */
```

```
  LONG     yTop;     /*  Y-coordinate of top edge of rectangle. */
} RECTL;

typedef RECTL *PRECTL;
```

RECTL Field - xLeft

xLeft (LONG)
X-coordinate of left-hand edge of rectangle.

RECTL Field - yBottom

yBottom (LONG)
Y-coordinate of bottom edge of rectangle.

RECTL Field - xRight

xRight (LONG)
X-coordinate of right-hand edge of rectangle.

RECTL Field - yTop

yTop (LONG)
Y-coordinate of top edge of rectangle.

SEARCH_INFO

Search information structure.

```
typedef struct _SEARCH_INFO {
  M_WPObject      *ClassToBeFound;               /*  Pointer to the object to be found. */
  BOOL             fSkipExtendedSearchCriteria;  /*  Flag to skip extended search criteria. */
  PVOID            pvoidExtendedCriteria;        /*  Pointer to extended criteria. */
} SEARCH_INFO;

typedef SEARCH_INFO *PSRCH_INFO;
```

SEARCH_INFO Field - ClassToBeFound

ClassToBeFound (M_WPObject *)
Pointer to the object to be found.

SEARCH_INFO Field - fSkipExtendedSearchCriteria

fSkipExtendedSearchCriteria (BOOL)
Flag to skip extended search criteria.

TRUE	Skip extended search criteria.
FALSE	Don't skip extended search criteria.

SEARCH_INFO Field - pvoidExtendedCriteria

pvoidExtendedCriteria (PVOID)
Pointer to extended criteria.

SGID

This data type is specific to versions, higher than version 3, of the OS/2 operating system.

Value used to hold a screen group ID.

```
#ifdef _PPC_
typedef ULONG SGID;
#else
typedef USHORT SGID;
#endif
```

SHORT

Signed integer in the range -32 768 through 32 767.

```
#define SHORT short
```

SIZEF

Size structure (FIXED values).

```
typedef struct _SIZEF {
  FIXED     cx;  /*  Width. */
  FIXED     cy;  /*  Height. */
} SIZEF;

typedef SIZEF *PSIZEF;
```

SIZEF Field - cx

cx (FIXED)
Width.

SIZEF Field - cy

cy (FIXED)
Height.

SIZEL

Size structure (LONG values).

```
typedef struct _SIZEL {
  LONG      cx;  /*  Width. */
  LONG      cy;  /*  Height. */
} SIZEL;

typedef SIZEL *PSIZEL;
```

SIZEL Field - cx

cx (LONG)
Width.

SIZEL Field - cy

cy (LONG)

Height.

SORTFASTINFO

Sort record data structure.

```
typedef struct _SORTFASTINFO {
  PFNCOMPARE       pfnCompare;   /*  Pointer to the details view comparison function. */
  ULONG            FieldOffset;
  M_WPObject      *Class;
} SORTFASTINFO;

typedef SORTFASTINFO *PSORTFASTINFO;
```

SORTFASTINFO Field - pfnCompare

pfnCompare (PFNCOMPARE)
Pointer to the details view comparison function.

SORTFASTINFO Field - FieldOffset

FieldOffset (ULONG)

SORTFASTINFO Field - Class

Class (M_WPObject *)

SWP

Set-window-position structure.

```
typedef struct _SWP {
  ULONG     fl;                /*  Options. */
  LONG      cy;                /*  Window height. */
  LONG      cx;                /*  Window width. */
  LONG      y;                 /*  Y-coordinate of origin. */
  LONG      x;                 /*  X-coordinate of origin. */
  HWND      hwndInsertBehind;  /*  Window behind which this window is placed. */
  HWND      hwnd;              /*  Window handle. */
  ULONG     ulReserved1;       /*  Reserved value; must be 0. */
```

```
  ULONG     ulReserved2;       /*  Reserved value; must be 0. */
} SWP;

typedef SWP *PSWP;
```

SWP Field - fl

fl (ULONG)
Options.

Possible values are shown in the following list:

SWP_ACTIVATE

SWP_DEACTIVATE

SWP_HIDE

SWP_MAXIMIZE

SWP_MINIMIZE

SWP_MOVE

SWP_NOADJUST

SWP_NOERASEWINDOW

SWP_NOREDRAW

SWP_RESTORE

SWP_SHOW

SWP_SIZE

SWP_ZORDER

SWP Field - cy

cy (LONG)
Window height.

SWP Field - cx

cx (LONG)
Window width.

SWP Field - y

y (LONG)
Y-coordinate of origin.

SWP Field - x

x (LONG)
X-coordinate of origin.

SWP Field - hwndInsertBehind

hwndInsertBehind (HWND)
Window behind which this window is placed.

SWP Field - hwnd

hwnd (HWND)
Window handle.

SWP Field - ulReserved1

ulReserved1 (ULONG)
Reserved value; must be 0.

SWP Field - ulReserved2

ulReserved2 (ULONG)
Reserved value; must be 0.

TASKREC

Task information structure.

```
typedef struct _TASKREC {
  struct _TASKREC      *next;                /*  Pointer to the next record in the list of current tasks. */
  ULONG                 useCount;            /*  Usage count (this structure can be referenced by multiple obje
  PVOID                 pStdDlg;             /*  Pointer to a standard Workplace Shell task dialog. */
  WPFolder             *folder;              /*  Pointer to the intended folder for the object. */
  LONG                  xOrigin;             /*  Intended x-coordinate position for the object. */
  LONG                  yOrigin;             /*  Intended y-coordinate position for the object. */
  PSZ                   pszTitle;            /*  Intended name for the object. */
  ULONG                 cbTitle;             /*  Length of the name buffer for pszTitle. */
  PMINIRECORDCORE       positionAfterRecord; /*  Intended sort position for the object. */
  BOOL                  fKeepAssociations;
  PVOID                 pReserved;           /*  Reserved.  Set to NULL. */
} TASKREC;

typedef TASKREC *PTASKREC;
```

TASKREC Field - next

next (struct _TASKREC *)
Pointer to the next record in the list of current tasks.

TASKREC Field - useCount

useCount (ULONG)
Usage count (this structure can be referenced by multiple objects).

TASKREC Field - pStdDlg

pStdDlg (PVOID)
Pointer to a standard Workplace Shell task dialog.

TASKREC Field - folder

folder (WPFolder *)
Pointer to the intended folder for the object.

TASKREC Field - xOrigin

xOrigin (LONG)

Intended x-coordinate position for the object.

TASKREC Field - yOrigin

yOrigin (LONG)
Intended y-coordinate position for the object.

TASKREC Field - pszTitle

pszTitle (PSZ)
Intended name for the object.

TASKREC Field - cbTitle

cbTitle (ULONG)
Length of the name buffer for *pszTitle* .

TASKREC Field - positionAfterRecord

positionAfterRecord (PMINIRECORDCORE)
Intended sort position for the object.

TASKREC Field - fKeepAssociations

fKeepAssociations (BOOL)

TASKREC Field - pReserved

pReserved (PVOID)
Reserved. Set to NULL.

TREEITEMDESC

Structure that contains icons and bit maps used to represent the state of an expanded or collapsed parent item in the tree name view of a container control.

```
typedef struct _TREEITEMDESC {
  HBITMAP      hbmExpanded;    /*  Expanded bit-map handle. */
  HBITMAP      hbmCollapsed;   /*  Collapsed bit-map handle. */
  HPOINTER     hptrExpanded;   /*  Expanded icon handle. */
  HPOINTER     hptrCollapsed;  /*  Collapsed icon handle. */
} TREEITEMDESC;

typedef TREEITEMDESC *PTREEITEMDESC;
```

TREEITEMDESC Field - hbmExpanded

hbmExpanded (HBITMAP)
Expanded bit-map handle.

The handle of the bit map to be used to represent an expanded parent item in the tree name view.

TREEITEMDESC Field - hbmCollapsed

hbmCollapsed (HBITMAP)
Collapsed bit-map handle.

The handle of the bit map to be used to represent a collapsed parent item in the tree name view.

TREEITEMDESC Field - hptrExpanded

hptrExpanded (HPOINTER)
Expanded icon handle.

The handle of the icon to be used to represent an expanded parent item in the tree name view.

TREEITEMDESC Field - hptrCollapsed

hptrCollapsed (HPOINTER)
Collapsed icon handle.

The handle of the icon to be used to represent a collapsed parent item in the tree name view.

UCHAR

Single-byte unsigned character or unsigned integer in the range 0 through 255.

```
typedef unsigned char UCHAR;
```

ULONG

32-bit unsigned integer in the range 0 through 4 294 967 295.

```
typedef unsigned long ULONG;
```

USEITEM

Use-item data structure.

The USEITEM structure is always followed by a type-specific structure that is indicated by the *type* field.

```
typedef struct _USEITEM {
  ULONG                 type;   /*  The type of this item. */
  struct _USEITEM      *pNext;  /*  Next item in the Use list. */
} USEITEM;

typedef USEITEM *PUSEITEM;
```

USEITEM Field - type

type (ULONG)
The type of this item.

Possible values are:

USAGE_LINK
LINKITEM
USAGE_MEMORY
MEMORYITEM
USAGE_NOTIFY
LINKITEM
USAGE_OPENVIEW
VIEWITEM
USAGE_OPENFILE
VIEWFILE
USAGE_RECORD
RECORDITEM

USEITEM Field - pNext

pNext (struct _USEITEM *)
Next item in the Use list.

USHORT

Unsigned integer in the range 0 through 65 535.

```
typedef unsigned short USHORT;
```

VIEWFILE

USAGE_OPENFILE structure.

```
typedef struct _VIEWFILE {
  ULONG               ulMenuId;  /*  Menu ID, if an association or menu page. */
  LHANDLE             handle;    /*  Open handle. */
  HWND                hwndCnr;   /*  System use only (window animation). */
  PMINIRECORDCORE     pRecord;   /*  System use only (window animation). */
} VIEWFILE;

typedef VIEWFILE *PVIEWFILE;
```

VIEWFILE Field - ulMenuId

ulMenuId (ULONG)
Menu ID, if an association or menu page.

VIEWFILE Field - handle

handle (LHANDLE)
Open handle.

VIEWFILE Field - hwndCnr

hwndCnr (HWND)
System use only (window animation).

VIEWFILE Field - pRecord

pRecord (PMINIRECORDCORE)
System use only (window animation).

VIEWITEM

USAGE_OPENVIEW structure.

```
typedef struct _VIEWITEM {
  ULONG               view;         /*  Object view that this represents. */
  LHANDLE             handle;       /*  Open handle. */
  ULONG               ulViewState;  /*  View state flags. */
  HWND                hwndCnr;      /*  System use only (window animation). */
  PMINIRECORDCORE     pRecord;      /*  System use only (window animation). */
} VIEWITEM;

typedef VIEWITEM *PVIEWITEM;
```

VIEWITEM Field - view

view (ULONG)
Object view that this represents.

VIEWITEM Field - handle

handle (LHANDLE)
Open handle.

VIEWITEM Field - ulViewState

ulViewState (ULONG)
View state flags.

Possible values are:

VIEWSTATE_CLOSING
The view is closing.
VIEWSTATE_DELETING
The object is being deleted.
VIEWSTATE_OBJECTDELETED

The object has been deleted.

VIEWSTATE_OPENING
: The view is being opened.

VIEWSTATE_PAINTED
: The view was painted.

VIEWSTATE_POPULATING
: The view is populating.

VIEWSTATE_USERHIDDEN
: The view was hidden by the user.

VIEWSTATE_USERMINIMIZED
: The view was minimized by the user.

VIEWITEM Field - hwndCnr

hwndCnr (HWND)
: **System use only** (window animation).

VIEWITEM Field - pRecord

pRecord (PMINIRECORDCORE)
: **System use only** (window animation).

VOID

A data area of undefined format.

```
#define VOID void
```

WPSRCLASSBLOCK

Save or restore the class block structure.

```
typedef struct _WPSRCLASSBLOCK {
  SHORT      ClassNameLength;  /*  Length of class name. */
  USHORT     IVarLength;       /*  Length of the instance variable information. */
} WPSRCLASSBLOCK;

typedef WPSRCLASSBLOCK *PWPSRCLASSBLOCK*;
```

WPSRCLASSBLOCK Field - ClassNameLength

ClassNameLength (SHORT)
Length of class name.

Includes the null terminator. This must be a short and must be at the beginning of the structure. The class name immediately follows the control block. The first instance variable control block immediately follows this.

WPSRCLASSBLOCK Field - IVarLength

IVarLength (USHORT)
Length of the instance variable information.

Includes the two-byte null terminator.

Menu IDs

The following Menu IDs are for popup menus.

WPMENUID_ACCESSNEW
WPMENUID_ARRANGE
WPMENUID_ASSIGN
WPMENUID_CHKDSK
WPMENUID_CLOSE
WPMENUID_COPY
WPMENUID_COPYDSK
WPMENUID_CREATEANOTHER
WPMENUID_CREATESHADOW
WPMENUID_DELETE
WPMENUID_DESELALL
WPMENUID_DETAILS
WPMENUID_EJECTDISK
WPMENUID_EXTENDEDHELP
WPMENUID_FIND
WPMENUID_FIXDSK
WPMENUID_FORMAT
WPMENUID_HELP
WPMENUID_HELP_FOR_HELP
WPMENUID_HELPINDEX
WPMENUID_HELPKEYS
WPMENUID_HOWTOGETHELP
WPMENUID_ICON
WPMENUID_LOCKDISK
WPMENUID_LOCKUP
WPMENUID_LOGIN
WPMENUID_LOGOUT
WPMENUID_MOVE
WPMENUID_OPEN
WPMENUID_OPENPARENT
WPMENUID_PALETTE
WPMENUID_PICKUP
WPMENUID_PRIMARY
WPMENUID_PRINT
WPMENUID_PROGRAM
WPMENUID_PROPERTIES
WPMENUID_PUTDOWN
WPMENUID_PUTDOWN_CANCEL
WPMENUID_PUTDOWN_COPY
WPMENUID_PUTDOWN_CREATE
WPMENUID_PUTDOWN_CREATEPROGRAMOBJECT
WPMENUID_PUTDOWN_LINK
WPMENUID_PUTDOWN_MOVE
WPMENUID_REFRESH
WPMENUID_SELECT

WPMENUID_SELALL
WPMENUID_SHUTDOWN
WPMENUID_SORT
WPMENUID_SYSTEMSETUP
WPMENUID_TREE
WPMENUID_UNASSIGN
WPMENUID_UNLOCKDISK

The following menu IDs are for the WPFolder menu bar.

WPMENUID_ARRANGEBOTTOM
WPMENUID_ARRANGELEFT
WPMENUID_ARRANGERIGHT
WPMENUID_ARRANGETOP
WPMENUID_CHANGETODETAILS
WPMENUID_CHANGETOICON
WPMENUID_CHANGETOTREE
WPMENUID_DOCSHELL
WPMENUID_EDITPULLDOWN
WPMENUID_FOLDERPULLDOWN
WPMENUID_HELPPULLDOWN
WPMENUID_LOCKEDINPLACE
WPMENUID_LOCKEDINPLACENO
WPMENUID_LOCKEDINPLACEYES
WPMENUID_LOGOFF
WPMENUID_MAXIMIZE
WPMENUID_PASTE
WPMENUID_PERIMETER
WPMENUID_RESTORE
WPMENUID_SELECTEDHORZ
WPMENUID_SELECTEDPULLDOWN
WPMENUID_SELECTEDVERT
WPMENUID_STANDARD
WPMENUID_UNDOARRANGE
WPMENUID_VIEW
WPMENUID_VIEWPULLDOWN
WPMENUID_WINDOWLIST

The following menu IDs are for the WPPort context menu.

WPPOIDM_PORT_INSTALL
WPPOIDM_PORT_REDIRECT

The following menu IDs are for the WPPrinter context menu.

WPPOIDM_PD_DEFAULT
WPPOIDM_PD_DELETE_JOBS
WPPOIDM_PD_DETAIL
WPPOIDM_PD_HOLD
WPPOIDM_PD_ICON
WPPOIDM_PD_INSTALL
WPPOICM_PD_MENU
WPPOICM_PD_OPENMENU
WPPOIDM_PD_RELEASE
WPPOIDM_PD_REFRESH
WPPOIDM_PD_STATUS

The following menu IDs are for the WPQdr context menu.

WPPOIDM_QDR_INSTALL

The following menu IDs are for the WPRPRinter context menu.

WPPOICM_RPRTMENU
WPPOIDM_RPRTACCESSANOTHER
WPPOIDM_RPRTCONNECT
WPPOIDM_RPRTDISCONNECT
WPPOIDM_RPRTLOGON
WPPOIDM_RPRTLOGOFF
WPPOIDM_RPRTREMADMIN
WPPOIDM_RPRTREMADMIN_COPY

WPPOIDM_RPRTREMADMIN_CREATE
WPPOIDM_RPRTREMADMIN_DELETE
WPPOIDM_PPRTINSTALL

The following menu IDs are for the WPSpool context menu.

WPPOICM_SPOOL_ENABLE 5001
WPPOIDM_SPOOL_ENABLE 5101
WPPOICM_SPOOL_DISABLE 5002
WPPOIDM_SPOOL_DISABLE 5102

All user-defined menu IDs should be greater than WPMENUID_USER.

Notices

(November 1996)

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Requests for technical information about IBM products should be made to your IBM reseller or IBM marketing representative.

Copyright Notices

COPYRIGHT LICENSE: This publication contains printed sample application programs in source language, which illustrate OS/2 programming techniques. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the OS/2 application programming interface.

Each copy of any portion of these sample programs or any derivative work, which is distributed to others, must include a copyright notice as follows: "(C) (your company name) (year). All rights reserved."

(C) Copyright International Business Machines Corporation 1994, 1996. All rights reserved.
Note to U.S. Government Users - Documentation related to restricted rights - Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Disclaimers

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation

500 Columbus Avenue
Thornwood, NY 10594
U.S.A.

Asia-Pacific users can inquire, in writing, to the IBM Director of Intellectual Property and Licensing, IBM World Trade Asia Corporation, 2-31 Roppongi 3-chome, Minato-ku, Tokyo 106, Japan.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Department LZKS, 11400 Burnet Road, Austin, TX 78758 U.S.A. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX
CUA
Information Presentation Facility
Operating System/2
PM
Proprinter
System Application Architecture
Workplace Shell
Common User Access
IBM
IPF
OS/2
Presentation Manager
SAA
WIN-OS/2

The following terms are trademarks of other companies:

C++
Helvetica
Microsoft Corporation
Novell, Inc.
Windows
American Telephone and Telegraph Company
Linotype Company
Novell
PCMCIA
Personal Computer Memory Card International Association

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Glossary

This glossary defines many of the terms used in this book. It includes terms and definitions from the *IBM Dictionary of Computing* , as well as terms specific to the OS/2 operating system and the Presentation Manager. It is not a complete glossary for the entire OS/2 operating system; nor is it a complete dictionary of computer terms.

Other primary sources for these definitions are:

- The *American National Standard Dictionary for Information Systems* , ANSI X3.172-1990, copyrighted 1990 by the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. These definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary* , developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.

Glossary Listing

Select a starting letter of glossary terms:

A
B
C
N
O
P

D Q
E R
F S
G T
H U
I V
J W
K X
L Y
M Z

Glossary - A

accelerator -In SAA Common User Access architecture, a key or combination of keys that invokes an application-defined function.

accelerator table -A table used to define which key strokes are treated as *accelerators* and the commands they are translated into.

access mode -The manner in which an application gains access to a file it has opened. Examples of access modes are read-only, write-only, and read/write.

access permission -All access rights that a user has regarding an object. (I)

action -One of a set of defined tasks that a computer performs. Users request the application to perform an action in several ways, such as typing a command, pressing a function key, or selecting the action name from an action bar or menu.

action bar -In SAA Common User Access architecture, the area at the top of a window that contains choices that give a user access to actions available in that window.

action point -The current position on the screen at which the pointer is pointing. Contrast with *hot spot* and *input focus* .

active program -A program currently running on the computer. An active program can be interactive (running and receiving input from the user) or noninteractive (running but not receiving input from the user). See also *interactive program* and *noninteractive program* .

active window -The window with which the user is currently interacting.

address space -(1) The range of addresses available to a program. (A) (2) The area of virtual storage available for a particular job.

alphanumeric video output -Output to the logical video buffer when the video adapter is in text mode and the logical video buffer is addressed by an application as a rectangular array of character cells.

American National Standard Code for Information Interchange -The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), that is used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters. (A)

Note: IBM has defined an extension to ASCII code (characters 128-255).

anchor -A window procedure that handles Presentation Manager message conversions between an icon procedure and an application.

anchor block -An area of Presentation-Manager-internal resources to allocated process or thread that calls WinInitialize.

anchor point -A point in a window used by a program designer or by a window manager to position a subsequently appearing window.

ANSI -American National Standards Institute.

APA -All points addressable.

API -Application programming interface.

application -A collection of software components used to perform specific types of work on a computer; for example, a payroll application, an airline reservation application, a network application.

application object -In SAA Advanced Common User Access architecture, a form that an application provides for a user; for example, a spreadsheet form. Contrast with *user object* .

application programming interface (API) -A functional interface supplied by the operating system or by a separately orderable licensed program that allows an application program written in a high-level language to use specific data or functions of the operating system or the licensed program.

application-modal -Pertaining to a message box or dialog box for which processing must be completed before further interaction with any

other window owned by the same application may take place.

area -In computer graphics, a filled shape such as a solid rectangle.

ASCII -American National Standard Code for Information Interchange.

ASCIIZ -A string of ASCII characters that is terminated with a byte containing the value 0.

aspect ratio -In computer graphics, the width-to-height ratio of an area, symbol, or shape.

asynchronous (ASYNC) -(1) Pertaining to two or more processes that do not depend upon the occurrence of specific events such as common timing signals. (T) (2) Without regular time relationship; unexpected or unpredictable with respect to the execution of program instructions. See also *synchronous* .

atom -A constant that represents a string. As soon as a string has been defined as an atom, the atom can be used in place of the string to save space. Strings are associated with their respective atoms in an *atom table* . See also *integer atom* .

atom table -A table used to relate *atoms* with the strings that they represent. Also in the table is the mechanism by which the presence of a string can be checked.

atomic operation -An operation that completes its work on an object before another operation can be performed on the same object.

attribute -A characteristic or property that can be controlled, usually to obtain a required appearance; for example, the color of a line. See also *graphics attributes* and *segment attributes* .

automatic link -In Information Presentation Facility (IPF), a link that begins a chain reaction at the primary window. When the user selects the primary window, an automatic link is activated to display secondary windows.

AVIO -Advanced Video Input/Output.

Glossary - B

Bézier curve -(1) A mathematical technique of specifying smooth continuous lines and surfaces, which require a starting point and a finishing point with several intermediate points that influence or control the path of the linking curve. Named after Dr. P. Bézier. (2) (D of C) In the AIX Graphics Library, a cubic spline approximation to a set of four control points that passes through the first and fourth control points and that has a continuous slope where two spline segments meet. Named after Dr. P. Bézier.

background -(1) In multiprogramming, the conditions under which low-priority programs are executed. Contrast with *foreground* . (2) An active session that is not currently displayed on the screen.

background color -The color in which the background of a graphic primitive is drawn.

background mix -An attribute that determines how the background of a graphic primitive is combined with the existing color of the graphics presentation space. Contrast with *mix* .

background program -In multiprogramming, a program that executes with a low priority. Contrast with *foreground program* .

bit map -A representation in memory of the data displayed on an APA device, usually the screen.

block -(1) A string of data elements recorded or transmitted as a unit. The elements may be characters, words, or logical records. (T) (2) To record data in a block. (3) A collection of contiguous records recorded as a unit. Blocks are separated by interblock gaps and each block may contain one or more records. (A)

block device -A storage device that performs I/O operations on blocks of data called *sectors* . Data on block devices can be randomly accessed. Block devices are designated by a drive letter (for example, **C:**).

blocking mode -A condition set by an application that determines when its threads might block. For example, an application might set the Pipemode parameter for the DosCreateNPipe function so that its threads perform I/O operations to the named pipe block when no data is available.

border -A visual indication (for example, a separator line or a background color) of the boundaries of a window.

boundary determination -An operation used to compute the size of the smallest rectangle that encloses a graphics object on the screen.

breakpoint -(1) A point in a computer program where execution may be halted. A breakpoint is usually at the beginning of an instruction where halts, caused by external intervention, are convenient for resuming execution. (T) (2) A place in a program, specified by a command or a condition, where the system halts execution and gives control to the workstation user or to a specified program.

broken pipe -When all of the handles that access one end of a pipe have been closed.

bucket -One or more fields in which the result of an operation is kept.

buffer -(1) A portion of storage used to hold input or output data temporarily. (2) To allocate and schedule the use of buffers. (A)

button -A mechanism used to request or initiate an action. See also *barrel buttons* , *bezel buttons* , *mouse button* , *push button* , and *radio button* .

byte pipe -Pipes that handle data as byte streams. All unnamed pipes are byte pipes. Named pipes can be byte pipes or message pipes. See *byte stream* .

byte stream -Data that consists of an unbroken stream of bytes.

Glossary - C

cache -A high-speed buffer storage that contains frequently accessed instructions and data; it is used to reduce access time.

cached micro presentation space -A presentation space from a Presentation-Manager-owned store of micro presentation spaces. It can be used for drawing to a window only, and must be returned to the store when the task is complete.

CAD -Computer-Aided Design.

call -(1) The action of bringing a computer program, a routine, or a subroutine into effect, usually by specifying the entry conditions and jumping to an entry point. (I) (A) (2) To transfer control to a procedure, program, routine, or subroutine.

calling sequence -A sequence of instructions together with any associated data necessary to execute a call. (T)

Cancel -An action that removes the current window or menu without processing it, and returns the previous window.

cascaded menu -In the OS/2 operating system, a menu that appears when the arrow to the right of a cascading choice is selected. It contains a set of choices that are related to the cascading choice. Cascaded menus are used to reduce the length of a menu. See also *cascading choice* .

cascading choice -In SAA Common User Access architecture, a choice in a menu that, when selected, produces a cascaded menu containing other choices. An arrow () appears to the right of the cascading choice.

CASE statement -In PM programming, provides the body of a window procedure. There is usually one CASE statement for each message type supported by an application.

CGA -Color graphics adapter.

chained list -A list in which the data elements may be dispersed but in which each data element contains information for locating the next. (T) Synonymous with *linked list* .

character -A letter, digit, or other symbol.

character box -In computer graphics, the boundary that defines, in world coordinates, the horizontal and vertical space occupied by a single character from a character set. See also *character mode* . Contrast with *character cell* .

character cell -The physical, rectangular space in which any single character is displayed on a screen or printer device. Position is addressed by row and column coordinates. Contrast with *character box* .

character code -The means of addressing a character in a character set, sometimes called *code point* .

character device -A device that performs I/O operations on one character at a time. Because character devices view data as a stream of bytes, character-device data cannot be randomly accessed. Character devices include the keyboard, mouse, and printer, and are referred to by name.

character mode -A mode that, in conjunction with the font type, determines the extent to which graphics characters are affected by the character box, shear, and angle attributes.

character set -(1) An ordered set of unique representations called characters; for example, the 26 letters of English alphabet, Boolean 0 and 1, the set of symbols in the Morse code, and the 128 ASCII characters. (A) (2) All the valid characters for a programming language or for a computer system. (3) A group of characters used for a specific reason; for example, the set of characters a printer can print.

check box -In SAA Advanced Common User Access architecture, a square box with associated text that represents a choice. When a user selects a choice, an X appears in the check box to indicate that the choice is in effect. The user can clear the check box by selecting the choice again. Contrast with *radio button* .

check mark - (1) (D of C) In SAA Advanced Common User Access architecture, a symbol that shows that a choice is currently in effect. (2) The symbol that is used to indicate a selected item on a pull-down menu.

child process -In the OS/2 operating system, a process started by another process, which is called the parent process. Contrast with *parent process* .

child window -A window that appears within the border of its parent window (either a primary window or another child window). When the parent window is resized, moved, or destroyed, the child window also is resized, moved, or destroyed; however, the child window can be moved or resized independently from the parent window, within the boundaries of the parent window. Contrast with *parent window* .

choice -(1) An option that can be selected. The choice can be presented as text, as a symbol (number or letter), or as an icon (a pictorial symbol). (2) (D of C) In SAA Common User Access architecture, an item that a user can select.

chord -(1) To press more than one button on a pointing device while the pointer is within the limits that the user has specified for the operating environment. (2) (D of C) In graphics, a short line segment whose end points lie on a circle. Chords are a means for producing a circular image from straight lines. The higher the number of chords per circle, the smoother the circular image.

class -A way of categorizing objects based on their behavior and shape. A class is, in effect, a definition of a generic object. In SOM, a class is a special kind of object that can manufacture other objects that all have a common shape and exhibit similar behavior (more precisely, all of the objects manufactured by a class have the same memory layout and share a common set of methods). New classes can be defined in terms of existing classes through a technique known as *inheritance* .

class method -A class method of class <X> is a method provided by the metaclass of class <X>. Class methods are executed without requiring any instances of class <X> to exist, and are frequently used to create instances. In System Object Model, an action that can be performed on a class object.

class object -In System Object Model, the run-time implementation of a class.

class style -The set of properties that apply to every window in a window class.

client -(1) A functional unit that receives shared services from a server. (T) (2) A user, as in a client process that uses a named pipe or queue that is created and owned by a server process.

client area -The part of the window, inside the border, that is below the menu bar. It is the user's work space, where a user types information and selects choices from selection fields. In primary windows, it is where an application programmer presents the objects that a user works on.

client program -An application that creates and manipulates instances of classes.

client window -The window in which the application displays output and receives input. This window is located inside the frame window, under the window title bar and any menu bar, and within any scroll bars.

clip limits -The area of the paper that can be reached by a printer or plotter.

clipboard -In SAA Common User Access architecture, an area of computer memory, or storage, that temporarily holds data. Data in the clipboard is available to other applications.

clipping -In computer graphics, removing those parts of a display image that lie outside a given boundary. (I) (A)

clipping area -The area in which the window can paint.

clipping path -A clipping boundary in world-coordinate space.

clock tick -The minimum unit of time that the system tracks. If the system timer currently counts at a rate of X Hz, the system tracks the time every 1/X of a second. Also known as *time tick* .

CLOCK$ -Character-device name reserved for the system clock.

code page -An assignment of graphic characters and control-function meanings to all code points.

code point -(1) Synonym for *character code* . (2) (D of C) A 1-byte code representing one of 256 potential characters.

code segment -An executable section of programming code within a load module.

color dithering -See *dithering* .

color graphics adapter (CGA) -An adapter that simultaneously provides four colors and is supported by all IBM Personal Computer and Personal System/2 models.

command -The name and parameters associated with an action that a program can perform.

command area -An area composed of a command field prompt and a command entry field.

command entry field -An entry field in which users type commands.

command line -On a display screen, a display line, sometimes at the bottom of the screen, in which only commands can be entered.

command mode -A state of a system or device in which the user can enter commands.

command prompt -A field prompt showing the location of the command entry field in a panel.

Common Programming Interface (CPI) -Definitions of those application development languages and services that have, or are intended to have, implementations on and a high degree of commonality across the SAA environments. One of the three SAA architectural areas. See also *Common User Access architecture* .

Common User Access (CUA) architecture - Guidelines for the dialog between a human and a workstation or terminal. One of the three SAA architectural areas. See also *Common Programming Interface* .

compile -To translate a program written in a higher-level programming language into a machine language program.

composite window -A window composed of other windows (such as a frame window, frame-control windows, and a client window) that are kept together as a unit and that interact with each other.

computer-aided design (CAD) -The use of a computer to design or change a product, tool, or machine, such as using a computer for drafting or illustrating.

COM1, COM2, COM3 -Character-device names reserved for serial ports 1 through 3.

CON -Character-device name reserved for the console keyboard and screen.

conditional cascaded menu -A pull-down menu associated with a menu item that has a cascade mini-push button beside it in an object's pop-up menu. The conditional cascaded menu is displayed when the user selects the mini-push button.

container -In SAA Common User Access architecture, an object that holds other objects. A folder is an example of a container object. See also *folder* and *object* .

contextual help -In SAA Common User Access Architecture, help that gives specific information about the item the cursor is on. The help is contextual because it provides information about a specific item as it is currently being used. Contrast with *extended help* .

contiguous -Touching or joining at a common edge or boundary, for example, an unbroken consecutive series of storage locations.

control -In SAA Advanced Common User Access architecture, a component of the user interface that allows a user to select choices or type information; for example, a check box, an entry field, a radio button.

control area -A storage area used by a computer program to hold control information. (I) (A)

Control Panel -In the Presentation Manager, a program used to set up user preferences that act globally across the system.

Control Program -(1) The basic functions of the operating system, including DOS emulation and the support for keyboard, mouse, and video input/output. (2) A computer program designed to schedule and to supervise the execution of programs of a computer system. (I) (A)

control window -A window that is used as part of a composite window to perform simple input and output tasks. Radio buttons and check boxes are examples.

control word -An instruction within a document that identifies its parts or indicates how to format the document.

coordinate space -A two-dimensional set of points used to generate output on a video display of printer.

Copy -A choice that places onto the clipboard, a copy of what the user has selected. See also *Cut* and *Paste* .

correlation -The action of determining which element or object within a picture is at a given position on the display. This follows a *pick* operation.

coverpage window -A window in which the application's help information is displayed.

CPI -Common Programming Interface.

critical extended attribute -An extended attribute that is necessary for the correct operation of the system or a particular application.

critical section -(1) In programming languages, a part of an asynchronous procedure that cannot be executed simultaneously with a certain part of another asynchronous procedure. (I)

Note: Part of the other asynchronous procedure also is a critical section. (2) A section of code that is not reentrant; that is, code that can be executed by only one thread at a time.

CUA architecture -Common User Access architecture.

current position -In computer graphics, the position, in user coordinates, that becomes the starting point for the next graphics routine, if that routine does not explicitly specify a starting point.

cursor -A symbol displayed on the screen and associated with an input device. The cursor indicates where input from the device will be placed. Types of cursors include text cursors, graphics cursors, and selection cursors. Contrast with *pointer* and *input focus* .

Cut -In SAA Common User Access architecture, a choice that removes a selected object, or a part of an object, to the clipboard, usually compressing the space it occupied in a window. See also *Copy* and *Paste* .

Glossary - D

daisy chain -A method of device interconnection for determining interrupt priority by connecting the interrupt sources serially.

data segment -A nonexecutable section of a program module; that is, a section of a program that contains data definitions.

data structure -The syntactic structure of symbolic expressions and their storage-allocation characteristics. (T)

data transfer -The movement of data from one object to another by way of the clipboard or by direct manipulation.

DBCS -Double-byte character set.

DDE -Dynamic data exchange.

deadlock -(1) Unresolved contention for the use of a resource. (2) An error condition in which processing cannot continue because each of two elements of the process is waiting for an action by, or a response from, the other. (3) An impasse that occurs when multiple processes are waiting for the availability of a resource that will not become available because it is being held by another process that is in a similar wait state.

debug -To detect, diagnose, and eliminate errors in programs. (T)

decipoint -In printing, one tenth of a point. There are 72 points in an inch.

default procedure -A function provided by the Presentation Manager Interface that may be used to process standard messages from dialogs or windows.

default value -A value assumed when no value has been specified. Synonymous with assumed value. For example, in the graphics programming interface, the default line-type is 'solid'.

definition list -A type of list that pairs a term and its description.

delta -An application-defined threshold, or number of container items, from either end of the list.

descendant -See *child process* .

descriptive text -Text used in addition to a field prompt to give more information about a field.

Deselect all -A choice that cancels the selection of all of the objects that have been selected in that window.

Desktop Manager -In the Presentation Manager, a window that displays a list of groups of programs, each of which can be started or stopped.

desktop window -The window, corresponding to the physical device, against which all other types of windows are established.

detached process -A background process that runs independent of the parent process.

detent -A point on a slider that represents an exact value to which a user can move the slider arm.

device context -A logical description of a data destination such as memory, metafile, display, printer, or plotter. See also *direct device* context, *information device context* , *memory device context* , *metafile device context* , *queued device context* , and *screen device context* .

device driver -A file that contains the code needed to attach and use a device such as a display, printer, or plotter.

device space -(1) Coordinate space in which graphics are assembled after all GPI transformations have been applied. Device space is defined in device-specific units. (2) (D of C) In computer graphics, a space defined by the complete set of addressable points of a display device. (A)

dialog -The interchange of information between a computer and its user through a sequence of requests by the user and the presentation of responses by the computer.

dialog box -In SAA Advanced Common User Access architecture, a movable window, fixed in size, containing controls that a user uses to provide information required by an application so that it can continue to process a user request. See also *message box, primary window, secondary window* . Also known as a *pop-up window* .

Dialog Box Editor -A *WYSIWYG* editor that creates dialog boxes for communicating with the application user.

dialog item -A component (for example, a menu or a button) of a dialog box. Dialog items are also used when creating dialog templates.

dialog procedure -A dialog window that is controlled by a window procedure. It is responsible for responding to all messages sent to the dialog window.

dialog tag language -A markup language used by the DTL compiler to create dialog objects.

dialog template -The definition of a dialog box, which contains details of its position, appearance, and window ID, and the window ID of each of its child windows.

direct device context -A logical description of a data destination that is a device other than the screen (for example, a printer or plotter), and where the output is not to go through the spooler. Its purpose is to satisfy queries. See also *device context* .

direct manipulation -The user's ability to interact with an object by using the mouse, typically by dragging an object around on the Desktop and dropping it on other objects.

direct memory access (DMA) -A technique for moving data directly between main storage and peripheral equipment without requiring processing of the data by the processing unit.(T)

directory -A type of file containing the names and controlling information for other files or other directories.

display point -Synonym for *pel* .

dithering -(1) The process used in color displays whereby every other pel is set to one color, and the intermediate pels are set to another. Together they produce the effect of a third color at normal viewing distances. This process can only be used on solid areas of color; it does not work, for example, on narrow lines. (2) (D of C) In computer graphics, a technique of interleaving dark and light pixels so that the resulting image looks smoothly shaded when viewed from a distance.

DMA -Direct memory access.

DOS Protect Mode Interface (DPMI) -An interface between protect mode and real mode programs.

double-byte character set (DBCS) -A set of characters in which each character is represented by two bytes. Languages such as Japanese, Chinese, and Korean, which contain more characters than can be represented by 256 code points, require double-byte character sets. Since each character requires two bytes, the entering, displaying, and printing of DBCS characters requires hardware and software that can support DBCS.

doubleword -A contiguous sequence of bits or characters that comprises two computer words and is capable of being addressed as a unit. (A)

DPMI -DOS Protect Mode Interface.

drag -In SAA Common User Access, to use a pointing device to move an object; for example, clicking on a window border, and dragging it to make the window larger.

dragging -(1) In computer graphics, moving an object on the display screen as if it were attached to the pointer. (2) (D of C) In computer graphics, moving one or more segments on a display surface by translating. (I) (A)

drawing chain -See *segment chain* .

drop -To fix the position of an object that is being dragged, by releasing the select button of the pointing device. See also *drag* .

DTL -Dialog tag language.

dual-boot function -A feature of the OS/2 operating system that allows the user to start DOS from within the operating system, or an OS/2 session from within DOS.

duplex -Pertaining to communication in which data can be sent and received at the same time. Synonymous with *full duplex* .

dynamic data exchange (DDE) -A message protocol used to communicate between applications that share data. The protocol uses shared memory as the means of exchanging data between applications.

dynamic data formatting -A formatting procedure that enables you to incorporate text, bit maps or metafiles in an IPF window at execution time.

dynamic link library -A collection of executable programming code and data that is bound to an application at load time or run time, rather than during linking. The programming code and data in a dynamic link library can be shared by several applications simultaneously.

dynamic linking -The process of resolving external references in a program module at load time or run time rather than during linking.

dynamic segments -Graphics segments drawn in exclusive-OR mix mode so that they can be moved from one screen position to another without affecting the rest of the displayed picture.

dynamic storage -(1) A device that stores data in a manner that permits the data to move or vary with time such that the specified data is not

always available for recovery. (A) (2) A storage in which the cells require repetitive application of control signals in order to retain stored data. Such repetitive application of the control signals is called a refresh operation. A dynamic storage may use static addressing or sensing circuits. (A) (3) See also *static storage* .

dynamic time slicing -Varies the size of the time slice depending on system load and paging activity.

dynamic-link module -A module that is linked at load time or run time.

Glossary - E

EBCDIC -Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters (9 bits including parity check), used for information interchange among data processing systems, data communications systems, and associated equipment.

edge-triggered -Pertaining to an event semaphore that is posted then reset before a waiting thread gets a chance to run. The semaphore is considered to be posted for the rest of that thread's waiting period; the thread does not have to wait for the semaphore to be posted again.

EGA -Extended graphics adapter.

element -An entry in a graphics segment that comprises one or more graphics orders and that is addressed by the element pointer.

EMS -Expanded Memory Specification.

encapsulation -Hiding an object's implementation, that is, its private, internal data and methods. Private variables and methods are accessible only to the object that contains them.

entry field -In SAA Common User Access architecture, an area where a user types information. Its boundaries are usually indicated. See also *selection field* .

entry panel -A defined panel type containing one or more entry fields and protected information such as headings, prompts, and explanatory text.

entry-field control -The component of a user interface that provides the means by which the application receives data entered by the user in an entry field. When it has the input focus, the entry field displays a flashing pointer at the position where the next typed character will go.

environment segment -The list of environment variables and their values for a process.

environment strings -ASCII text strings that define the value of environment variables.

environment variables -Variables that describe the execution environment of a process. These variables are named by the operating system or by the application. Environment variables named by the operating system are PATH, DPATH, INCLUDE, INIT, LIB, PROMPT, and TEMP. The values of environment variables are defined by the user in the CONFIG.SYS file, or by using the SET command at the OS/2 command prompt.

error message -An indication that an error has been detected. (A)

event semaphore -A semaphore that enables a thread to signal a waiting thread or threads that an event has occurred or that a task has been completed. The waiting threads can then perform an action that is dependent on the completion of the signaled event.

exception -An abnormal condition such as an I/O error encountered in processing a data set or a file.

exclusive system semaphore -A system semaphore that can be modified only by threads within the same process.

executable file -(1) A file that contains programs or commands that perform operations or actions to be taken. (2) A collection of related data records that execute programs.

exit -To execute an instruction within a portion of a computer program in order to terminate the execution of that portion. Such portions of computer programs include loops, subroutines, modules, and so on. (T) Repeated exit requests return the user to the point from which all functions provided to the system are accessible. Contrast with *cancel* .

expanded memory specification (EMS) -Enables DOS applications to access memory above the 1MB real mode addressing limit.

extended attribute -An additional piece of information about a file object, such as its data format or category. It consists of a name and a value. A file object may have more than one extended attribute associated with it.

extended help -In SAA Common User Access architecture, a help action that provides information about the contents of the application window from which a user requested help. Contrast with *contextual help* .

extended-choice selection -A mode that allows the user to select more than one item from a window. Not all windows allow extended choice

selection. Contrast with *multiple-choice selection* .

extent -Continuous space on a disk or diskette that is occupied by or reserved for a particular data set, data space, or file.

external link -In Information Presentation Facility, a link that connects external online document files.

Glossary - F

family-mode application -An application program that can run in the OS/2 environment and in the DOS environment; however, it cannot take advantage of many of the OS/2-mode facilities, such as multitasking, interprocess communication, and dynamic linking.

FAT -File allocation table.

FEA -Full extended attribute.

field-level help -Information specific to the field on which the cursor is positioned. This help function is "contextual" because it provides information about a specific item as it is currently used; the information is dependent upon the context within the work session.

FIFO -First-in-first-out. (A)

file -A named set of records stored or processed as a unit. (T)

file allocation table (FAT) -In IBM personal computers, a table used by the operating system to allocate space on a disk for a file, and to locate and chain together parts of the file that may be scattered on different sectors so that the file can be used in a random or sequential manner.

file attribute -Any of the attributes that describe the characteristics of a file.

File Manager -In the Presentation Manager, a program that displays directories and files, and allows various actions on them.

file specification -The full identifier for a file, which includes its drive designation, path, file name, and extension.

file system -The combination of software and hardware that supports storing information on a storage device.

file system driver (FSD) -A program that manages file I\O and controls the format of information on the storage media.

fillet -A curve that is tangential to the end points of two adjoining lines. See also *polyfillet* .

filtering -An application process that changes the order of data in a queue.

first-in-first-out (FIFO) -A queuing technique in which the next item to be retrieved is the item that has been in the queue for the longest time. (A)

flag -(1) An indicator or parameter that shows the setting of a switch. (2) A character that signals the occurrence of some condition, such as the end of a word. (A) (3) (D of C) A characteristic of a file or directory that enables it to be used in certain ways. See also *archive flag* , *hidden flag* , and *read-only flag* .

focus -See *input focus* .

folder -A container used to organize objects.

font -A particular size and style of typeface that contains definitions of character sets, marker sets, and pattern sets.

Font Editor -A utility program provided with the IBM Developers Toolkit that enables the design and creation of new fonts.

foreground program -(1) The program with which the user is currently interacting. Also known as *interactive program* . Contrast with *background program* . (2) (D of C) In multiprogramming, a high-priority program.

frame -The part of a window that can contain several different visual elements specified by the application, but drawn and controlled by the Presentation Manager. The frame encloses the client area.

frame styles -Standard window layouts provided by the Presentation Manager.

FSD -File system driver.

full-duplex -Synonym for *duplex* .

full-screen application -An application that has complete control of the screen.

function -(1) In a programming language, a block, with or without formal parameters, whose execution is invoked by means of a call. (2) A set of related control statements that cause one or more programs to be performed.

function key -A key that causes a specified sequence of operations to be performed when it is pressed, for example, F1 and Alt-K.

function key area -The area at the bottom of a window that contains function key assignments such as F1=Help.

Glossary - G

GDT -Global Descriptor Table.

general protection fault -An exception condition that occurs when a process attempts to use storage or a module that has some level of protection assigned to it, such as I/O privilege level. See also *IOPL code segment* .

Global Descriptor Table (GDT) -A table that defines code and data segments available to all tasks in an application.

global dynamic-link module -A dynamic-link module that can be shared by all processes in the system that refer to the module name.

global file-name character -Either a question mark (?) or an asterisk (*) used as a variable in a file name or file name extension when referring to a particular file or group of files.

glyph -A graphic symbol whose appearance conveys information.

GPI -Graphics programming interface.

graphic primitive -In computer graphics, a basic element, such as an arc or a line, that is not made up of smaller parts and that is used to create diagrams and pictures. See also *graphics segment* .

graphics -(1) A picture defined in terms of graphic primitives and graphics attributes. (2) (D of C) The making of charts and pictures. (3) Pertaining to charts, tables, and their creation. (4) See *computer graphics, coordinate graphics, fixed-image graphics, interactive graphics, passive graphics, raster graphics* .

graphics attributes -Attributes that apply to graphic primitives. Examples are color, line type, and shading-pattern definition. See also *segment attributes* .

graphics field -The clipping boundary that defines the visible part of the presentation-page contents.

graphics mode -One of several states of a display. The mode determines the resolution and color content of the screen.

graphics model space -The conceptual coordinate space in which a picture is constructed after any model transforms have been applied. Also known as *model space* .

Graphics programming interface -The formally defined programming language that is between an IBM graphics program and the user of the program.

graphics segment -A sequence of related graphic primitives and graphics attributes. See also *graphic primitive* .

graying -The indication that a choice on a pull-down is unavailable.

group -A collection of logically connected controls. For example, the buttons controlling paper size for a printer could be called a group. See also *program group* .

Glossary - H

handle -(1) An identifier that represents an object, such as a device or window, to the Presentation Interface. (2) (D of C) In the Advanced DOS and OS/2 operating systems, a binary value created by the system that identifies a drive, directory, and file so that the file can be found and opened.

hard error -An error condition on a network that requires either that the system be reconfigured or that the source of the error be removed before the system can resume reliable operation.

header -(1) System-defined control information that precedes user data. (2) The portion of a message that contains control information for the

message, such as one or more destination fields, name of the originating station, input sequence number, character string indicating the type of message, and priority level for the message.

heading tags -A document element that enables information to be displayed in windows, and that controls entries in the contents window controls placement of push buttons in a window, and defines the shape and size of windows.

heap -An area of free storage available for dynamic allocation by an application. Its size varies according to the storage requirements of the application.

help function -(1) A function that provides information about a specific field, an application panel, or information about the help facility. (2) (D of C) One or more display images that describe how to use application software or how to do a system operation.

Help index -In SAA Common User Access architecture, a help action that provides an index of the help information available for an application.

help panel -A panel with information to assist users that is displayed in response to a help request from the user.

help window -A Common-User-Access-defined secondary window that displays information when the user requests help.

hidden file -An operating system file that is not displayed by a directory listing.

hide button -In the OS/2 operating system, a small, square button located in the right-hand corner of the title bar of a window that, when selected, removes from the screen all the windows associated with that window. Contrast with *maximize button* . See also *restore button* .

hierarchical inheritance -The relationship between parent and child classes. An object that is lower in the inheritance hierarchy than another object, inherits all the characteristics and behaviors of the objects above it in the hierarchy.

hierarchy -A tree of segments beginning with the root segment and proceeding downward to dependent segment types.

high-performance file system (HPFS) -In the OS/2 operating system, an installable file system that uses high-speed buffer storage, known as a cache, to provide fast access to large disk volumes. The file system also supports the coexistence of multiple, active file systems on a single personal computer, with the capability of multiple and different storage devices. File names used with the HPFS can have as many as 254 characters.

hit testing -The means of identifying which window is associated with which input device event.

hook -A point in a system-defined function where an application can supply additional code that the system processes as though it were part of the function.

hook chain -A sequence of hook procedures that are "chained" together so that each event is passed, in turn, to each procedure in the chain.

hot spot -The part of the pointer that must touch an object before it can be selected. This is usually the tip of the pointer. Contrast with *action point* .

HPFS -high-performance file system.

hypergraphic link -A connection between one piece of information and another through the use of graphics.

hypertext -A way of presenting information online with connections between one piece of information and another, called *hypertext links* . See also *hypertext link* .

hypertext link -A connection between one piece of information and another.

Glossary - I

I/O operation -An input operation to, or output operation from a device attached to a computer.

I-beam pointer -A pointer that indicates an area, such as an entry field in which text can be edited.

icon -In SAA Advanced Common User Access architecture, a graphical representation of an object, consisting of an image, image background, and a label. Icons can represent items (such as a document file) that the user wants to work on, and actions that the user wants to perform. In the Presentation Manager, icons are used for data objects, system actions, and minimized programs.

icon area -In the Presentation Manager, the area at the bottom of the screen that is normally used to display the icons for minimized programs.

Icon Editor -The Presentation Manager-provided tool for creating icons.

IDL -Interface Definition Language.

image font -A set of symbols, each of which is described in a rectangular array of pels. Some of the pels in the array are set to produce the image of one of the symbols. Contrast with *outline font* .

implied metaclass -Subclassing the metaclass of a parent class without a separate CSC for the resultant metaclass.

indirect manipulation -Interaction with an object through choices and controls.

information device context -A logical description of a data destination other than the screen (for example, a printer or plotter), but where no output will occur. Its purpose is to satisfy queries. See also *device context* .

information panel -A defined panel type characterized by a body containing only protected information.

Information Presentation Facility (IPF) -A facility provided by the OS/2 operating system, by which application developers can produce online documentation and context-sensitive online help panels for their applications.

inheritance -The technique of specifying the shape and behavior of one class (called a *subclass*) as incremental differences from another class (called the *parent class* or *superclass*). The subclass inherits the superclass' state representation and methods, and can provide additional data elements and methods. The subclass also can provide new functions with the same method names used by the superclass. Such a subclass method is said to override the superclass method, and will be selected automatically by method resolution on subclass instances. An overriding method can elect to call upon the superclass' method as part of its own implementation.

input focus -(1) The area of a window where user interaction is possible using an input device, such as a mouse or the keyboard. (2) The position in the *active window* where a user's normal interaction with the keyboard will appear.

input router -An internal OS/2 process that removes messages from the system queue.

input/output control -A device-specific command that requests a function of a device driver.

installable file system (IFS) -A file system in which software is installed when the operating system is started.

instance -(Or object instance). A specific object, as distinguished from the abstract definition of an object referred to as its class.

instance method -A method valid for a particular object.

instruction pointer -In System/38, a pointer that provides addressability for a machine interface instruction in a program.

integer atom -An *atom* that represents a predefined system constant and carries no storage overhead. For example, names of window classes provided by Presentation Manager are expressed as integer atoms.

interactive graphics -Graphics that can be moved or manipulated by a user at a terminal.

interactive program -(1) A program that is running (active) and is ready to receive (or is receiving) input from a user. (2) A running program that can receive input from the keyboard or another input device. Compare with *active program* and contrast with *noninteractive* program.

Also known as a *foreground program* .

interchange file -A file containing data that can be sent from one Presentation Manager interface application to another.

Interface Definition Language (IDL) -Language-neutral interface specification for a SOM class.

interpreter -A program that translates and executes each instruction of a high-level programming language before it translates and executes.

interprocess communication (IPC) -In the OS/2 operating system, the exchange of information between processes or threads through semaphores, pipes, queues, and shared memory.

interval timer -(1) A timer that provides program interruptions on a program-controlled basis. (2) An electronic counter that counts intervals of time under program control.

IOCtl -Input/output control.

IOPL -Input/output privilege level.

IOPL code segment -An IOPL executable section of programming code that enables an application to directly manipulate hardware interrupts and ports without replacing the device driver. See also *privilege level* .

IPC -Interprocess communication.

IPF -Information Presentation Facility.

IPF compiler -A text compiler that interpret tags in a source file and converts the information into the specified format.

IPF tag language -A markup language that provides the instructions for displaying online information.

item -A data object that can be passed in a DDE transaction.

Glossary - J

journal -A special-purpose file that is used to record changes made in the system.

Glossary - K

Kanji -A graphic character set used in Japanese ideographic alphabets.

KBD$ -Character-device name reserved for the keyboard.

kernel -The part of an operating system that performs basic functions, such as allocating hardware resources.

kerning -The design of graphics characters so that their character boxes overlap. Used to space text proportionally.

keyboard accelerator -A keystroke that generates a command message for an application.

keyboard augmentation -A function that enables a user to press a keyboard key while pressing a mouse button.

keyboard focus -A temporary attribute of a window. The window that has a keyboard focus receives all keyboard input until the focus changes to a different window.

Keys help -In SAA Common User Access architecture, a help action that provides a listing of the application keys and their assigned functions.

Glossary - L

label -In a graphics segment, an identifier of one or more elements that is used when editing the segment.

LAN -Local area network.

language support procedure -A function provided by the Presentation Manager Interface for applications that do not, or cannot (as in the case of COBOL and FORTRAN programs), provide their own dialog or window procedures.

lazy drag -See *pickup and drop* .

lazy drag set -See *pickup set* .

LDT -In the OS/2 operating system, Local Descriptor Table.

LIFO stack -A stack from which data is retrieved in last-in, first-out order.

linear address -A unique value that identifies the memory object.

linked list -Synonym for *chained list* .

list box -In SAA Advanced Common User Access architecture, a control that contains scrollable choices from which a user can select one choice.

Note: In CUA architecture, this is a programmer term. The end user term is selection list.

list button -A button labeled with an underlined down-arrow that presents a list of valid objects or choices that can be selected for that field.

list panel -A defined panel type that displays a list of items from which users can select one or more choices and then specify one or more actions to work on those choices.

load time -The point in time at which a program module is loaded into main storage for execution.

load-on-call -A function of a linkage editor that allows selected segments of the module to be disk resident while other segments are executing. Disk resident segments are loaded for execution and given control when any entry point that they contain is called.

local area network (LAN) -(1) A computer network located on a user's premises within a limited geographical area. Communication within a local area network is not subject to external regulations; however, communication across the LAN boundary may be subject to some form of regulation. (T)

Note: A LAN does not use store and forward techniques. (2) A network in which a set of devices are connected to one another for communication and that can be connected to a larger network.

Local Descriptor Table (LDT) -Defines code and data segments specific to a single task.

lock -A serialization mechanism by means of which a resource is restricted for use by the holder of the lock.

logical storage device -A device that the user can map to a physical (actual) device.

LPT1, LPT2, LPT3 -Character-device names reserved for parallel printers 1 through 3.

Glossary - M

main window -The window that is positioned relative to the *desktop window* .

manipulation button -The button on a pointing device a user presses to directly manipulate an object.

map -(1) A set of values having a defined correspondence with the quantities or values of another set. (I) (A) (2) To establish a set of values having a defined correspondence with the quantities or values of another set. (I)

marker box -In computer graphics, the boundary that defines, in world coordinates, the horizontal and vertical space occupied by a single marker from a marker set.

marker symbol -A symbol centered on a point. Graphs and charts can use marker symbols to indicate the plotted points.

marquee box -The rectangle that appears during a selection technique in which a user selects objects by drawing a box around them with a pointing device.

Master Help Index -In the OS/2 operating system, an alphabetic list of help topics related to using the operating system.

maximize -To enlarge a window to its largest possible size.

media window -The part of the physical device (display, printer, or plotter) on which a picture is presented.

memory block -Part memory within a heap.

memory device context -A logical description of a data destination that is a memory bit map. See also *device context* .

memory management -A feature of the operating system for allocating, sharing, and freeing main storage.

memory object -Logical unit of memory requested by an application, which forms the granular unit of memory manipulation from the application viewpoint.

menu -In SAA Advanced Common User Access architecture, an extension of the menu bar that displays a list of choices available for a selected choice in the menu bar. After a user selects a choice in menu bar, the corresponding menu appears. Additional pop-up windows can appear from menu choices.

menu bar -In SAA Advanced Common User Access architecture, the area near the top of a window, below the title bar and above the rest of the window, that contains choices that provide access to other menus.

menu button -The button on a pointing device that a user presses to view a pop-up menu associated with an object.

message -(1) In the Presentation Manager, a packet of data used for communication between the Presentation Manager interface and Presentation Manager applications (2) In a user interface, information not requested by users but presented to users by the computer in response to a user action or internal process.

message box -(1) A dialog window predefined by the system and used as a simple interface for applications, without the necessity of creating dialog-template resources or dialog procedures. (2) (D of C) In SAA Advanced Common User Access architecture, a type of window that shows messages to users. See also *dialog box, primary window, secondary window* .

message filter -The means of selecting which messages from a specific window will be handled by the application.

message queue -A sequenced collection of messages to be read by the application.

message stream mode -A method of operation in which data is treated as a stream of messages. Contrast with *byte stream* .

metacharacter -See *global file-name character* .

metaclass -A class whose instances are all classes. In SOM, any class descended from SOMClass is a metaclass. The methods of a metaclass are sometimes called "class" methods.

metafile -A file containing a series of attributes that set color, shape and size, usually of a picture or a drawing. Using a program that can interpret these attributes, a user can view the assembled image.

metafile device context -A logical description of a data destination that is a metafile, which is used for graphics interchange. See also *device context* .

metalanguage -A language used to specify another language. For example, data types can be described using a metalanguage so as to make the descriptions independent of any one computer language.

method -One of the units that makes up the behavior of an object. A method is a combination of a function and a name, such that many different functions can have the same name. Which function the name refers to at any point in time depends on the object that is to execute the method and is the subject of method resolution.

method override -The replacement, by a child class, of the implementation of a method inherited from a parent and an ancestor class.

mickey -A unit of measurement for physical mouse motion whose value depends on the mouse device driver currently loaded.

micro presentation space -A graphics presentation space in which a restricted set of the GPI function calls is available.

minimize -To remove from the screen all windows associated with an application and replace them with an icon that represents the application.

mix -An attribute that determines how the foreground of a graphic primitive is combined with the existing color of graphics output. Also known as *foreground mix* . Contrast with *background mix* .

mixed character string -A string containing a mixture of one-byte and *Kanji* or Hangeul (two-byte) characters.

mnemonic -(1) A method of selecting an item on a pull-down by means of typing the highlighted letter in the menu item. (2) (D of C) In SAA Advanced Common User Access architecture, usually a single character, within the text of a choice, identified by an underscore beneath the character. If all characters in a choice already serve as mnemonics for other choices, another character, placed in parentheses immediately following the choice, can be used. When a user types the mnemonic for a choice, the choice is either selected or the cursor is moved to that choice.

modal dialog box -In SAA Advanced Common User Access architecture, a type of movable window, fixed in size, that requires a user to enter information before continuing to work in the application window from which it was displayed. Contrast with *modeless dialog box* . Also known as a *serial dialog box* . Contrast with *parallel dialog box* .

Note: In CUA architecture, this is a programmer term. The end user term is pop-up window.

model space -See *graphics model space* .

modeless dialog box -In SAA Advanced Common User Access architecture, a type of movable window, fixed in size, that allows users to continue their dialog with the application without entering information in the dialog box. Also known as a *parallel dialog box* . Contrast with *modal dialog box* .

Note: In CUA architecture, this is a programmer term. The end user term is pop-up window.

module definition file -A file that describes the code segments within a load module. For example, it indicates whether a code segment is loadable before module execution begins (preload), or loadable only when referred to at run time (load-on-call).

mouse -In SAA usage, a device that a user moves on a flat surface to position a pointer on the screen. It allows a user to select a choice o function to be performed or to perform operations on the screen, such as dragging or drawing lines from one position to another.

MOUSE$ -Character-device name reserved for a mouse.

multiple-choice selection -In SAA Basic Common User Access architecture, a type of field from which a user can select one or more choices or select none. See also *check box* . Contrast with *extended-choice selection* .

multiple-line entry field -In SAA Advanced Common User Access architecture, a control into which a user types more than one line of information. See also *single-line entry field* .

multitasking -The concurrent processing of applications or parts of applications. A running application and its data are protected from other concurrently running applications.

mutex semaphore -(Mutual exclusion semaphore). A semaphore that enables threads to serialize their access to resources. Only the thread that currently owns the mutex semaphore can gain access to the resource, thus preventing one thread from interrupting operations being performed by another.

muxwait semaphore -(Multiple wait semaphore). A semaphore that enables a thread to wait either for multiple event semaphores to be posted or for multiple mutex semaphores to be released. Alternatively, a muxwait semaphore can be set to enable a thread to wait for any ONE of the event or mutex semaphores in the muxwait semaphore's list to be posted or released.

Glossary - N

named pipe -A named buffer that provides client-to-server, server-to-client, or full duplex communication between unrelated processes. Contrast with *unnamed pipe* .

national language support (NLS) -The modification or conversion of a United States English product to conform to the requirements of another language or country. This can include the enabling or retrofitting of a product and the translation of nomenclature, MRI, or documentation of a product.

nested list -A list that is contained within another list.

NLS -national language support.

non-8.3 file-name format -A file-naming convention in which file names can consist of up to 255 characters. See also *8.3 file-name format* .

noncritical extended attribute -An extended attribute that is not necessary for the function of an application.

nondestructive read -Reading that does not erase the data in the source location. (T)

noninteractive program -A running program that cannot receive input from the keyboard or other input device. Compare with *active* program, and contrast with *interactive program* .

nonretained graphics -Graphic primitives that are not remembered by the Presentation Manager interface when they have been drawn. Contrast with *retained graphics* .

null character (NUL) -(1) Character-device name reserved for a nonexistent (dummy) device. (2) (D of C) A control character that is used to accomplish media-fill or time-fill and that may be inserted into or removed from a sequence of characters without affecting the meaning of the sequence; however, the control of equipment or the format may be affected by this character. (I) (A)

null-terminated string -A string of (n+1) characters where the (n+1)th character is the 'null' character (0x00) Also known as 'zero-terminated' string and 'ASCIIZ' string.

Glossary - O

object -The elements of data and function that programs create, manipulate, pass as arguments, and so forth. An object is a way of associating specific data values with a specific set of named functions (called *methods*) for a period of time (referred to as the *lifetime* of the object). The data values of an object are referred to as its *state* . In SOM, objects are created by other objects called *classes* . The specification of what comprises the set of functions and data elements that make up an object is referred to as the *definition* of a class.

SOM objects offer a high degree of *encapsulation* . This property permits many aspects of the implementation of an object to change without affecting client programs that depend on the object's behavior.

object definition -See *class* .

object instance -See *instance* .

Object Interface Definition Language (OIDL) -Specification language used in SOM Version 1 for defining classes. Replaced by Interface Definition Language (IDL).

object window -A window that does not have a parent but which might have child windows. An object window cannot be presented on a device.

OIDL -Object Interface Definition Language.

open -To start working with a file, directory, or other object.

ordered list -Vertical arrangements of items, with each item in the list preceded by a number or letter.

outline font -A set of symbols, each of which is created as a series of lines and curves. Synonymous with *vector font* . Contrast with *image font* .

output area -An area of storage reserved for output. (A)

owner window -A window into which specific events that occur in another (owned) window are reported.

ownership -The determination of how windows communicate using messages.

owning process -The process that owns the resources that might be shared with other processes.

Glossary - P

page -(1) A 4KB segment of contiguous physical memory. (2) (D of C) A defined unit of space on a storage medium.

page viewport -A boundary in device coordinates that defines the area of the output device in which graphics are to be displayed. The presentation-page contents are transformed automatically to the page viewport in device space.

paint -(1) The action of drawing or redrawing the contents of a window. (2) In computer graphics, to shade an area of a display image; for example, with crosshatching or color.

panel -In SAA Basic Common User Access architecture, a particular arrangement of information that is presented in a window or pop-up. If some of the information is not visible, a user can scroll through the information.

panel area -An area within a panel that contains related information. The three major Common User Access-defined panel areas are the action bar, the function key area, and the panel body.

panel area separator -In SAA Basic Common User Access architecture, a solid, dashed, or blank line that provides a visual distinction between two adjacent areas of a panel.

panel body -The portion of a panel not occupied by the action bar, function key area, title or scroll bars. The panel body can contain protected information, selection fields, and entry fields. The layout and content of the panel body determine the panel type.

panel body area -See *client area* .

panel definition -A description of the contents and characteristics of a panel. A panel definition is the application developer's mechanism for predefining the format to be presented to users in a window.

panel ID -In SAA Basic Common User Access architecture, a panel identifier, located in the upper-left corner of a panel. A user can choose whether to display the panel ID.

panel title -In SAA Basic Common User Access architecture, a particular arrangement of information that is presented in a window or pop-up. If some of the information is not visible, a user can scroll through the information.

paper size -The size of paper, defined in either standard U.S. or European names (for example, A, B, A4), and measured in inches or millimeters respectively.

parallel dialog box -See *modeless dialog box* .

parameter list -A list of values that provides a means of associating addressability of data defined in a called program with data in the calling program. It contains parameter names and the order in which they are to be associated in the calling and called program.

parent class -See *inheritance* .

parent process -In the OS/2 operating system, a process that creates other processes. Contrast with *child process* .

parent window -In the OS/2 operating system, a window that creates a child window. The child window is drawn within the parent window. If the parent window is moved, resized, or destroyed, the child window also will be moved, resized, or destroyed. However, the child window can be moved and resized independently from the parent window, within the boundaries of the parent window. Contrast with *child window* .

partition -(1) A fixed-size division of storage. (2) On an IBM personal computer fixed disk, one of four possible storage areas of variable size; one may be accessed by DOS, and each of the others may be assigned to another operating system.

Paste -A choice in the Edit pull-down that a user selects to move the contents of the clipboard into a preselected location. See also *Copy* and

Cut .

path -The route used to locate files; the storage location of a file. A fully qualified path lists the drive identifier, directory name, subdirectory name (if any), and file name with the associated extension.

PDD -Physical device driver.

peeking -An action taken by any thread in the process that owns the queue to examine queue elements without removing them.

pel -(1) The smallest area of a display screen capable of being addressed and switched between visible and invisible states. Synonym for *display point* , *pixel* , and *picture element* . (2) (D of C) Picture element.

persistent object -An object whose instance data and state are preserved between system shutdown and system startup.

physical device driver (PDD) -A system interface that handles hardware interrupts and supports a set of input and output functions.

pick -To select part of a displayed object using the pointer.

pickup -To add an object or set of objects to the pickup set.

pickup and drop -A drag operation that does not require the direct manipulation button to be pressed for the duration of the drag.

pickup set -The set of objects that have been picked up as part of a pickup and drop operation.

picture chain -See *segment chain* .

picture element -(1) Synonym for *pel* . (2) (D of C) In computer graphics, the smallest element of a display surface that can be independently assigned color and intensity. (T) . (3) The area of the finest detail that can be reproduced effectively on the recording medium.

PID -Process identification.

pipe -(1) A named or unnamed buffer used to pass data between processes. A process reads from or writes to a pipe as if the pipe were a standard-input or standard-output file. See also *named pipe* and *unnamed pipe* . (2) (D of C) To direct data so that the output from one process becomes the input to another process. The standard output of one command can be connected to the standard input of another with the pipe operator (|).

pixel -(1) Synonym for *pel* . (2) (D of C) Picture element.

plotter -An output unit that directly produces a hardcopy record of data on a removable medium, in the form of a two-dimensional graphic representation. (T)

PM -Presentation Manager.

pointer -(1) The symbol displayed on the screen that is moved by a pointing device, such as a *mouse* . The pointer is used to point at items that users can select. Contrast with *cursor* . (2) A data element that indicates the location of another data element. (T)

POINTER$ -Character-device name reserved for a pointer device (mouse screen support).

pointing device -In SAA Advanced Common User Access architecture, an instrument, such as a mouse, trackball, or joystick, used to move a pointer on the screen.

pointings -Pairs of x-y coordinates produced by an operator defining positions on a screen with a pointing device, such as a *mouse* .

polyfillet -A curve based on a sequence of lines. The curve is tangential to the end points of the first and last lines, and tangential also to the midpoints of all other lines. See also *fillet* .

polygon -One or more closed figures that can be drawn filled, outlined, or filled and outlined.

polyline -A sequence of adjoining lines.

polymorphism -The ability to have different implementations of the same method for two or more classes of objects.

pop -To retrieve an item from a last-in-first-out stack of items. Contrast with *push* .

pop-up menu -A menu that lists the actions that a user can perform on an object. The contents of the pop-up menu can vary depending on the context, or state, of the object.

pop-up window -(1) A window that appears on top of another window in a dialog. Each pop-up window must be completed before returning to the underlying window. (2) (D of C) In SAA Advanced Common User Access architecture, a movable window, fixed in size, in which a user provides information required by an application so that it can continue to process a user request.

presentation drivers -Special purpose I/O routines that handle field device-independent I/O requests from the PM and its applications.

Presentation Manager (PM) -The interface of the OS/2 operating system that presents, in windows a graphics-based interface to applications and files installed and running under the OS/2 operating system.

presentation page -The coordinate space in which a picture is assembled for display.

presentation space (PS) -(1) Contains the device-independent definition of a picture. (2) (D of C) The display space on a display device.

primary window -In SAA Common User Access architecture, the window in which the main interaction between the user and the application takes place. In a multiprogramming environment, each application starts in its own primary window. The primary window remains for the duration of the application, although the panel displayed will change as the user's dialog moves forward. See also *secondary window* .

primitive -In computer graphics, one of several simple functions for drawing on the screen, including, for example, the rectangle, line, ellipse, polygon, and so on.

primitive attribute -A specifiable characteristic of a graphic primitive. See *graphics attributes* .

print job -The result of sending a document or picture to be printed.

Print Manager -In the Presentation Manager, the part of the spooler that manages the spooling process. It also allows users to view print queues and to manipulate print jobs.

privilege level -A protection level imposed by the hardware architecture of the IBM personal computer. There are four privilege levels (number 0 through 3). Only certain types of programs are allowed to execute at each privilege level. See also *IOPL code segment* .

procedure call -In programming languages, a language construct for invoking execution of a procedure.

process -An instance of an executing application and the resources it is using.

program -A sequence of instructions that a computer can interpret and execute.

program details -Information about a program that is specified in the *Program Manager* window and is used when the program is started.

program group -In the Presentation Manager, several programs that can be acted upon as a single entity.

program name -The full file specification of a program. Contrast with *program title* .

program title -The name of a program as it is listed in the *Program Manager* window. Contrast with *program name* .

prompt -A displayed symbol or message that requests input from the user or gives operational information; for example, on the display screen of an IBM personal computer, the DOS A> prompt. The user must respond to the prompt in order to proceed.

protect mode -A method of program operation that limits or prevents access to certain instructions or areas of storage. Contrast with *real mode* .

protocol -A set of semantic and syntactic rules that determines the behavior of functional units in achieving communication. (I)

pseudocode -An artificial language used to describe computer program algorithms without using the syntax of any particular programming language. (A)

pull-down -(1) An *action bar* extension that displays a list of choices available for a selected action bar choice. After users select an action bar choice, the pull-down appears with the list of choices. Additional *pop-up windows* may appear from pull-down choices to further extend the actions available to users. (2) (D of C) In SAA Common User Access architecture, pertaining to a choice in an action bar pull-down.

push -To add an item to a last-in-first-out stack of items. Contrast with *pop* .

push button -In SAA Advanced Common User Access architecture, a rectangle with text inside. Push buttons are used in windows for actions that occur immediately when the push button is selected.

putback -To remove an object or set of objects from the lazy drag set. This has the effect of undoing the pickup operation for those objects

putdown -To drop the objects in the lazy drag set on the target object.

Glossary - Q

queue -(1) A linked list of elements waiting to be processed in FIFO order. For example, a queue may be a list of print jobs waiting to be printed. (2) (D of C) A line or list of items waiting to be processed; for example, work to be performed or messages to be displayed.

queued device context -A logical description of a data destination (for example, a printer or plotter) where the output is to go through the spooler. See also *device context* .

Glossary - R

radio button -(1) A control window, shaped like a round button on the screen, that can be in a checked or unchecked state. It is used to select a single item from a list. Contrast with *check box* . (2) In SAA Advanced Common User Access architecture, a circle with text beside it. Radio buttons are combined to show a user a fixed set of choices from which only one can be selected. The circle is partially filled when a choice is selected.

RAS -Reliability, availability, and serviceability.

raster -(1) In computer graphics, a predetermined pattern of lines that provides uniform coverage of a display space. (T) (2) The coordinate grid that divides the display area of a display device. (A)

read-only file -A file that can be read from but not written to.

real mode -A method of program operation that does not limit or prevent access to any instructions or areas of storage. The operating system loads the entire program into storage and gives the program access to all system resources. Contrast with *protect mode* .

realize -To cause the system to ensure, wherever possible, that the physical color table of a device is set to the closest possible match in the logical color table.

recursive routine -A routine that can call itself, or be called by another routine that was called by the recursive routine.

reentrant -The attribute of a program or routine that allows the same copy of the program or routine to be used concurrently by two or more tasks.

reference phrase -(1) A word or phrase that is emphasized in a device-dependent manner to inform the user that additional information for the word or phrase is available. (2) (D of C) In hypertext, text that is highlighted and preceded by a single-character input field used to signify the existence of a hypertext link.

reference phrase help -In SAA Common User Access architecture, highlighted words or phrases within help information that a user selects to get additional information.

refresh -To update a window, with changed information, to its current status.

region -A clipping boundary in device space.

register -A part of internal storage having a specified storage capacity and usually intended for a specific purpose. (T)

remote file system -A file-system driver that gains access to a remote system without a block device driver.

resource -The means of providing extra information used in the definition of a window. A resource can contain definitions of fonts, templates, accelerators, and mnemonics; the definitions are held in a resource file.

resource file -A file containing information used in the definition of a window. Definitions can be of fonts, templates, accelerators, and mnemonics.

restore -To return a window to its original size or position following a sizing or moving action.

retained graphics -Graphic primitives that are remembered by the Presentation Manager interface after they have been drawn. Contrast with *nonretained graphics* .

return code -(1) A value returned to a program to indicate the results of an operation requested by that program. (2) A code used to influence the execution of succeeding instructions.(A)

reverse video -(1) A form of highlighting a character, field, or cursor by reversing the color of the character, field, or cursor with its background; for example, changing a red character on a black background to a black character on a red background. (2) In SAA Basic Common User Access architecture, a screen emphasis feature that interchanges the foreground and background colors of an item.

REXX Language -Restructured Extended Executor. A procedural language that provides batch language functions along with structured programming constructs such as loops; conditional testing and subroutines.

RGB -(1) Color coding in which the brightness of the additive primary colors of light, red, green, and blue, are specified as three distinct values of white light. (2) Pertaining to a color display that accepts signals representing red, green, and blue.

roman -Relating to a type style with upright characters.

root segment -In a hierarchical database, the highest segment in the tree structure.

round-robin scheduling -A process that allows each thread to run for a specified amount of time.

run time -(1) Any instant at which the execution of a particular computer program takes place. (T) (2) The amount of time needed for the execution of a particular computer program. (T) (3) The time during which an instruction in an instruction register is decoded and performed. Synonym for *execution time* .

Glossary - S

SAA -Systems Application Architecture.

SBCS -Single-byte character set.

scheduler -A computer program designed to perform functions such as scheduling, initiation, and termination of jobs.

screen -In SAA Basic Common User Access architecture, the physical surface of a display device upon which information is shown to a user.

screen device context -A logical description of a data destination that is a particular window on the screen. See also *device context* .

SCREEN$ -Character-device name reserved for the display screen.

scroll bar -In SAA Advanced Common User Access architecture, a part of a window, associated with a scrollable area, that a user interacts with to see information that is not currently allows visible.

scrollable entry field -An entry field larger than the visible field.

scrollable selection field -A selection field that contains more choices than are visible.

scrolling -Moving a display image vertically or horizontally in a manner such that new data appears at one edge, as existing data disappears at the opposite edge.

secondary window -A window that contains information that is dependent on information in a primary window and is used to supplement the interaction in the primary window.

sector -On disk or diskette storage, an addressable subdivision of a track used to record one block of a program or data.

segment -See *graphics segment* .

segment attributes -Attributes that apply to the segment as an entity, as opposed to the individual primitives within the segment. For example, the visibility or detectability of a segment.

segment chain -All segments in a graphics presentation space that are defined with the 'chained' attribute. Synonym for *picture chain* .

segment priority -The order in which segments are drawn.

segment store -An area in a normal graphics presentation space where retained graphics segments are stored.

select -To mark or choose an item. Note that *select* means to mark or type in a choice on the screen; *enter* means to send all selected choices to the computer for processing.

select button -The button on a pointing device, such as a mouse, that is pressed to select a menu choice. Also known as button 1.

selection cursor -In SAA Advanced Common User Access architecture, a visual indication that a user has selected a choice. It is represented by outlining the choice with a dotted box. See also *text cursor* .

selection field -(1) In SAA Advanced Common User Access architecture, a set of related choices. See also *entry field* . (2) In SAA Basic Common User Access architecture, an area of a panel that cannot be scrolled and contains a fixed number of choices.

semantics -The relationships between symbols and their meanings.

semaphore -An object used by applications for signalling purposes and for controlling access to serially reusable resources.

separator -In SAA Advanced Common User Access architecture, a line or color boundary that provides a visual distinction between two adjacent areas.

serial dialog box -See *modal dialog box* .

serialization -The consecutive ordering of items.

serialize -To ensure that one or more events occur in a specified sequence.

serially reusable resource (SRR) -A logical resource or object that can be accessed by only one task at a time.

session -(1) A routing mechanism for user interaction via the console; a complete environment that determines how an application runs and how users interact with the application. OS/2 can manage more than one session at a time, and more than one process can run in a session. Each session has its own set of environment variables that determine where OS/2 looks for dynamic-link libraries and other important files. (2) (D of C) In the OS/2 operating system, one instance of a started program or command prompt. Each session is separate from all other sessions that might be running on the computer. The operating system is responsible for coordinating the resources that each session uses, such as computer memory, allocation of processor time, and windows on the screen.

Settings Notebook -A control window that is used to display the settings for an object and to enable the user to change them.

shadow -An object that refers to another object. A shadow is not a copy of another object, but is another representation of the object.

shadow box -The area on the screen that follows mouse movements and shows what shape the window will take if the mouse button is released.

shared data -Data that is used by two or more programs.

shared memory -In the OS/2 operating system, a segment that can be used by more than one program.

shear -In computer graphics, the forward or backward slant of a graphics symbol or string of such symbols relative to a line perpendicular to the baseline of the symbol.

shell -(1) A software interface between a user and the operating system of a computer. Shell programs interpret commands and user interactions on devices such as keyboards, pointing devices, and touch-sensitive screens, and communicate them to the operating system. (2) Software that allows a kernel program to run under different operating-system environments.

shutdown -The process of ending operation of a system or a subsystem, following a defined procedure.

sibling processes -Child processes that have the same parent process.

sibling windows -Child windows that have the same parent window.

simple list -A list of like values; for example, a list of user names. Contrast with *mixed list* .

single-byte character set (SBCS) -A character set in which each character is represented by a one-byte code. Contrast with *double-byte character set* .

slider box -In SAA Advanced Common User Access architecture: a part of the scroll bar that shows the position and size of the visible information in a window relative to the total amount of information available. Also known as *thumb mark* .

SOM -System Object Model.

source file -A file that contains source statements for items such as high-level language programs and data description specifications.

source statement -A statement written in a programming language.

specific dynamic-link module -A dynamic-link module created for the exclusive use of an application.

spin button -In SAA Advanced Common User Access architecture, a type of entry field that shows a scrollable ring of choices from which a user can select a choice. After the last choice is displayed, the first choice is displayed again. A user can also type a choice from the scrollable ring into the entry field without interacting with the spin button.

spline -A sequence of one or more Bézier curves.

spooler -A program that intercepts the data going to printer devices and writes it to disk. The data is printed or plotted when it is complete and the required device is available. The spooler prevents output from different sources from being intermixed.

stack -A list constructed and maintained so that the next data element to be retrieved is the most recently stored. This method is characterized as last-in-first-out (LIFO).

standard window -A collection of window elements that form a panel. The standard window can include one or more of the following window elements: sizing borders, system menu icon, title bar, maximize/minimize/restore icons, action bar and pull-downs, scroll bars, and client area.

static control -The means by which the application presents descriptive information (for example, headings and descriptors) to the user. The user cannot change this information.

static storage -(1) A read/write storage unit in which data is retained in the absence of control signals. (A) Static storage may use dynamic addressing or sensing circuits. (2) Storage other than *dynamic storage* . (A)

style -See *window style* .

subclass -A class that inherits from another class. See also *Inheritance* .

subdirectory -In an IBM personal computer, a file referred to in a root directory that contains the names of other files stored on the diskette or fixed disk.

superclass -A class from which another class inherits. See also *inheritance* .

swapping -(1) A process that interchanges the contents of an area of real storage with the contents of an area in auxiliary storage. (I) (A) (2) In a system with virtual storage, a paging technique that writes the active pages of a job to auxiliary storage and reads pages of another job from auxiliary storage into real storage. (3) The process of temporarily removing an active job from main storage, saving it on disk, and processing another job in the area of main storage formerly occupied by the first job.

switch -(1) In SAA usage, to move the cursor from one point of interest to another; for example, to move from one screen or window to another or from a place within a displayed image to another place on the same displayed image. (2) In a computer program, a conditional instruction and an indicator to be interrogated by that instruction. (3) A device or programming technique for making a selection, for example, a toggle, a conditional jump.

switch list -See *Task List* .

symbolic identifier -A text string that equates to an integer value in an include file, which is used to identify a programming object.

symbols -In Information Presentation Facility, a document element used to produce characters that cannot be entered from the keyboard.

synchronous -Pertaining to two or more processes that depend upon the occurrence of specific events such as common timing signals. (T) See also *asynchronous* .

System Menu -In the Presentation Manager, the pull-down in the top left corner of a window that allows it to be moved and sized with the keyboard.

System Object Model (SOM) -A mechanism for language-neutral, object-oriented programming in the OS/2 environment.

system queue -The master queue for all pointer device or keyboard events.

system-defined messages -Messages that control the operations of applications and provides input an other information for applications to process.

Systems Application Architecture (SAA) -A set of IBM software interfaces, conventions, and protocols that provide a framework for designing and developing applications that are consistent across systems.

Glossary - T

table tags -In Information Presentation Facility, a document element that formats text in an arrangement of rows and columns.

tag -(1) One or more characters attached to a set of data that contain information about the set, including its identification. (I) (A) (2) In Generalized Markup Language markup, a name for a type of document or document element that is entered in the source document to identify it.

target object -An object to which the user is transferring information.

Task List -In the Presentation Manager, the list of programs that are active. The list can be used to switch to a program and to stop programs.

terminate-and-stay-resident (TSR) -Pertaining to an application that modifies an operating system interrupt vector to point to its own location (known as hooking an interrupt).

text -Characters or symbols.

text cursor -A symbol displayed in an entry field that indicates where typed input will appear.

text window -Also known as the VIO window.

text-windowed application -The environment in which the operating system performs advanced-video input and output operations.

thread -A unit of execution within a process. It uses the resources of the process.

thumb mark -The portion of the scroll bar that describes the range and properties of the data that is currently visible in a window. Also known as a *slider box* .

thunk -Term used to describe the process of address conversion, stack and structure realignment, etc., necessary when passing control between 16-bit and 32-bit modules.

tilde -A mark used to denote the character that is to be used as a mnemonic when selecting text items within a menu.

time slice -(1) An interval of time on the processing unit allocated for use in performing a task. After the interval has expired, processing-unit

time is allocated to another task, so a task cannot monopolize processing-unit time beyond a fixed limit. (2) In systems with time sharing, a segment of time allocated to a terminal job.

time-critical process -A process that must be performed within a specified time after an event has occurred.

timer -A facility provided under the Presentation Manager, whereby Presentation Manager will dispatch a message of class WM_TIMER to a particular window at specified intervals. This capability may be used by an application to perform a specific processing task at predetermined intervals, without the necessity for the application to explicitly keep track of the passage of time.

timer tick -See *clock tick* .

title bar -In SAA Advanced Common User Access architecture, the area at the top of each window that contains the window title and system menu icon. When appropriate, it also contains the minimize, maximize, and restore icons. Contrast with *panel title* .

TLB -Translation lookaside buffer.

transaction -An exchange between a workstation and another device that accomplishes a particular action or result.

transform -(1) The action of modifying a picture by scaling, shearing, reflecting, rotating, or translating. (2) The object that performs or defines such a modification; also referred to as a *transformation* .

Translation lookaside buffer (TLB) -A hardware-based address caching mechanism for paging information.

Tree -In the Presentation Manager, the window in the *File Manager* that shows the organization of drives and directories.

truncate -(1) To terminate a computational process in accordance with some rule (A) (2) To remove the beginning or ending elements of a string. (3) To drop data that cannot be printed or displayed in the line width specified or available. (4) To shorten a field or statement to a specified length.

TSR -Terminate-and-stay-resident.

Glossary - U

unnamed pipe -A circular buffer, created in memory, used by related processes to communicate with one another. Contrast with *named* pipe.

unordered list -In Information Presentation Facility, a vertical arrangement of items in a list, with each item in the list preceded by a special character or bullet.

update region -A system-provided area of dynamic storage containing one or more (not necessarily contiguous) rectangular areas of a window that are visually invalid or incorrect, and therefore are in need of repainting.

user interface -Hardware, software, or both that allows a user to interact with and perform operations on a system, program, or device.

User Shell -A component of OS/2 that uses a graphics-based, windowed interface to allow the user to manage applications and files installed and running under OS/2.

utility program -(1) A computer program in general support of computer processes; for example, a diagnostic program, a trace program, a sort program. (T) (2) A program designed to perform an everyday task such as copying data from one storage device to another. (A)

Glossary - V

value set control -A visual component that enables a user to select one choice from a group of mutually exclusive choices.

vector font -A set of symbols, each of which is created as a series of lines and curves. Synonymous with *outline font* . Contrast with *image font* .

VGA -Video graphics array.

view -A way of looking at an object's information.

viewing pipeline -The series of transformations applied to a graphic object to map the object to the device on which it is to be presented.

viewing window -A clipping boundary that defines the visible part of model space.

VIO -Video Input/Output.

virtual memory (VM) -Synonymous with *virtual storage* .

virtual storage -(1) The storage space that may be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme of the computer system and by the amount of auxiliary storage available, not by the actual number of main storage locations. (I) (A) (2) Addressable space that is apparent to the user as the processor storage space, from which the instructions and the data are mapped into the processor storage locations. (3) Synonymous with *virtual memory* .

visible region -A window's presentation space, clipped to the boundary of the window and the boundaries of any overlying window.

volume -(1) A file-system driver that uses a block device driver for input and output operations to a local or remote device. (I) (2) A portion of data, together with its data carrier, that can be handled conveniently as a unit.

Glossary - W

wildcard character -Synonymous with *global file-name character* .

window -(1) A portion of a display surface in which display images pertaining to a particular application can be presented. Different applications can be displayed simultaneously in different windows. (A) (2) An area of the screen with visible boundaries within which information is displayed. A window can be smaller than or the same size as the screen. Windows can appear to overlap on the screen.

window class -The grouping of windows whose processing needs conform to the services provided by one window procedure.

window coordinates -A set of coordinates by which a window position or size is defined; measured in device units, or *pels* .

window handle -Unique identifier of a window, generated by Presentation Manager when the window is created, and used by applications to direct messages to the window.

window procedure -Code that is activated in response to a message. The procedure controls the appearance and behavior of its associated windows.

window rectangle -The means by which the size and position of a window is described in relation to the desktop window.

window resource -A read-only data segment stored in the .EXE file of an application o the .DLL file of a dynamic link library.

window style -The set of properties that influence how events related to a particular window will be processed.

window title -In SAA Advanced Common User Access architecture, the area in the title bar that contains the name of the application and the OS/2 operating system file name, if applicable.

Workplace Shell -The OS/2 object-oriented, graphical user interface.

workstation -(1) A display screen together with attachments such as a keyboard, a local copy device, or a tablet. (2) (D of C) One or more programmable or nonprogrammable devices that allow a user to do work.

world coordinates -A device-independent Cartesian coordinate system used by the application program for specifying graphical input and output. (I) (A)

world-coordinate space -Coordinate space in which graphics are defined before transformations are applied.

WYSIWYG -What-You-See-Is-What-You-Get. A capability of a text editor to continually display pages exactly as they will be printed.

Glossary - X

There are no glossary terms for this starting letter.

Glossary - Y

There are no glossary terms for this starting letter.

Glossary - Z

z-order -The order in which sibling windows are presented. The topmost sibling window obscures any portion of the siblings that it overlaps; the same effect occurs down through the order of lower sibling windows.

zooming -The progressive scaling of an entire display image in order to give the visual impression of movement of all or part of a display group toward or away from an observer. (I) (A)

8.3 file-name format -A file-naming convention in which file names are limited to eight characters before and three characters after a single dot. Usually pronounced "eight-dot-three." See also *non-8.3 file-name format* .
